# A Game-Theoretic Approach to Strategic Resource Allocation Mechanisms in Edge and Fog Computing

**Suleiman Onimisi Aliyu**

Faculty of Computing, Engineering and Media (CEM)

De Montfort University, Leicester

This dissertation is submitted for the degree of

*Doctor of Philosophy in Computer Science*

2021

I would like to dedicate this thesis to loving Dad & Mum…

# Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other University. This dissertation is the output of own work and includes nothing, which is the outcome of work done in collaboration, except where specifically indicated in the text. This dissertation contains less than 46,000 words including appendices, bibliography, footnotes, tables, and equations and has less than 40 figures.

Suleiman Onimisi Aliyu
2021

# Acknowledgement

I would like to acknowledge the efforts of supervisory team (Dr Feng Chen & Dr Ying He) for their unwavering support during study. Their guidance and instructions made experience much more productive and memorable.

I am grateful to loved ones for their support during this difficult time especially only sister, and two brothers.

I would also like to acknowledge the help of all colleagues in the department of computer science and informatics, De Montfort University, Leicester, UK. Last but not the least; special thanks go to both sponsors i.e., De Montfort University and Abubakar Tafawa Balewa University, Bauchi, Nigeria for providing financial support throughout studies. It was not possible without their support.

# List of Relevant Publications

1. Aliyu, S.O. and Sim, K.M., 2014. Minimizing Message Exchanges in Agent based Cloud Service Composition. In *International Multiconference of Engineers and Computer Scientist 2014* (Vol. 1, pp. 12-14).
2. Aliyu, S.O., Chen, F. and Li, H., 2016, August. A Self-Tuning procedure for resource management in InterCloud Computing. In *2016 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)* (pp. 326-333). IEEE.
3. Aliyu, S.O., Chen, F. and He, Y., 2017, July. QoS-aware resource management in SDN-based interclouds: A software cybernetics perspective. In *2017 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)* (pp. 429-434). IEEE.
4. Aliyu, S.O., Chen, F., He, Y. and Yang, H., 2017, July. A game-theoretic based QoS-aware capacity management for real-time Edge-IoT applications. In *2017 IEEE International Conference on Software Quality, Reliability and Security (QRS)* (pp. 386-397). IEEE.
5. Yang, H., Chen, F. and Aliyu, S. O., 2017. Modern software cybernetics: new trends. Journal of Systems and Software. Elsevier.

# List of Abbreviations/Terms

| | |
|---|---|
| **CCF** | **Constrained Coalition Formation** |
| **CEEI** | **Competitive Equilibrium Equal Incomes** |
| **CMC** | **controlled Markov chains** |
| **CPP** | **Controller Placement Problem** |
| **CSG** | **Coalition Structure Generation** |
| **DC** | **Data centre** |
| **DRF** | **Dominant Resource Fairness** |
| **DRFH** | **Dominant Resource Fairness for Heterogenous servers** |
| **EC** | **Edge-Cloud** |
| **FEC** | **Fog and Edge Computing** |
| **FSM** | **Finite State Machines** |
| **GT** | **Game Theory** |
| **ICG** | **Inter-edge Coalition Games** |
| **IL** | **Independent Learners** |
| **IoT** | **Internet of Things** |
| **IP** | **Integer Partition** |
| **JAL** | **Joint Action Learners** |
| **MAS** | **Multi-Agent Systems** |

| | |
|---|---|
| **MARL** | **Multi-Agent Reinforcement Learning** |
| **MIP** | **Mixed Integer Programming** |
| **NFV** | **Network function virtualisation** |
| **NP** | **Non-deterministic Polynomial** |
| **PFG** | **Partition Form Game** |
| **PO** | **Pareto Optimal** |
| **QoS** | **Quality of Service** |
| **RAM** | **Resource Allocation Mechanism** |
| **RL** | **Reinforcement Learning** |
| **SBPO** | **Second-best Pareto Optimal** |
| **SDC** | **Software Defined Cloud Computing** |
| **SDN** | **Software Defined Networking** |
| **SLO** | **Service Level Objectives** |
| **VCG** | **Vickrey-Clarke-Groove Mechanism** |
| **VNF** | **Virtual Network Functions** |
| **WMN** | **Wireless Mesh Network** |

# Abstract

With the rapid growth of Internet of Things (IoT), cloud-centric application management raises questions related to quality of service for real-time applications. Fog and edge computing (FEC) provide a complement to the cloud by filling the gap between cloud and IoT. Resource management on multiple resources from distributed and administrative FEC nodes is a key challenge to ensure the quality of end-user's experience. To improve resource utilisation and system performance, researchers have been proposed many fair allocation mechanisms for resource management. Dominant Resource Fairness (DRF), a resource allocation policy for multiple resource types, meets most of the required fair allocation characteristics. However, DRF is suitable for centralised resource allocation without considering the effects (or feedbacks) of large-scale distributed environments like multi-controller software defined networking (SDN). Nash bargaining from micro-economic theory or competitive equilibrium equal incomes (CEEI) are well suited to solving dynamic optimisation problems proposing to 'proportionately' share resources among distributed participants. Although CEEI's decentralised policy guarantees *load balancing* for performance isolation, they are not fault-proof for computation offloading.

The thesis aims to propose a hybrid and fair allocation mechanism for *rejuvenation of decentralised SDN controller deployment*. We apply multi-agent reinforcement learning (MARL) with robustness against adversarial controllers to enable efficient priority scheduling for FEC. Motivated by software cybernetics and homeostasis, weighted DRF is generalised by applying the principles of feedback (positive or/and negative network effects) in reverse game theory (GT) to design hybrid scheduling schemes for joint multi-resource and multitask offloading/forwarding in FEC environments.

In the first piece of study, monotonic scheduling for joint offloading at the federated edge is addressed by proposing truthful mechanism (algorithmic) to neutralise harmful negative and positive distributive bargain externalities respectively. The IP-DRF scheme is a MARL approach applying partition form game (PFG) to guarantee second-best Pareto optimality

(SBPO) in allocation of multi-resources from deterministic policy in both population and resource non-monotonicity settings. In the second study, we propose DFog-DRF scheme to address truthful fog scheduling with bottleneck fairness in fault-probable wireless hierarchical networks by applying constrained coalition formation (CCF) games to implement MARL. The multi-objective optimisation problem for fog throughput maximisation is solved via a constraint dimensionality reduction methodology using fairness constraints for efficient gateway and low-level controller's placement.

For evaluation, we develop an agent-based framework to implement fair allocation policies in distributed data centre environments. In empirical results, the deterministic policy of IP-DRF scheme provides SBPO and reduces the average execution and turnaround time by 19% and 11.52% as compared to the Nash bargaining or CEEI deterministic policy for 57,445 cloudlets in population non-monotonic settings. The processing cost of tasks shows significant improvement (6.89% and 9.03% for fixed and variable pricing) for the resource non-monotonic setting - using 38,000 cloudlets. The DFog-DRF scheme when benchmarked against asset fair (MIP) policy shows superior performance (less than 1% in time complexity) for up to 30 FEC nodes. Furthermore, empirical results using 210 mobiles and 420 applications prove the efficacy of our hybrid scheduling scheme for hierarchical clustering considering latency and network usage for throughput maximisation.

# Table of Contents

# List of Figures

# List of Tables

# Chapter 1 Introduction

## 1.1 Background & motivation

Within the evolution of related technologies and the Internet, we have seen the emergence of a new paradigm, the Internet of Things (IoT). Essentially, IoT is viewed as a network of physical objects, such as devices, embedded with electronics, software, sensors, and network connectivity that enables these objects to collect and exchange data. In the IoT paradigm, many objects are interconnected for information sharing, regardless of their locations [1]. The concept of IoT has been nurtured for decades, even though the term IoT was first used by Kevin Ashton of Procter & Gamble (P&G) in 1999. As the realisation of ideas and technologies for the IoT evolve, different conceptual views for the implementation of the IoT emerge [2].

In the opinion of many experts, IoT is seen as a type of futuristic technology, which implies machine-to-machine communications and people to computer communications. Indeed, almost all categories of devices will become an IoT device, and they would occupy the role of consumers (or data providers), such as, streetlights, air quality sensors and even an Internet-enabled microwave oven. Typically, Cloud-centric IoT is the de facto architecture characterised by physical objects in the form of web resources that are managed by data centres or servers in the global internet. It would be safe to predict that the number of IoT at the network edge will grow to over billions in coming years. Therefore, data generated by them will be enormous, subverting the ability of Cloud-centric IoT system to deal with all the data because of bandwidth, latency, resource-constraint etc. Most of the data produced by IoT would probably never be transmitted to the remote cloud; instead, it will be consumed at the edge of the network.

Cloud computing is an intrinsic part of next-generation network service backend for applications on connected devices. With advances in Software defined Networking (SDN) and Network Function Virtualisation (NFV), the notion of virtualisation in the cloud data centre has extended to all resources – including storage, compute, and networks. *Fog and Edge computing (FEC)* are coined to complement the remote cloud to meet offloading demands of geographically distributed large number of IoT. The concept of ***Software Defined Clouds*** (SDC) aims to utilise the advances in cloud computing and virtualisation, SDN and NFV, to enhance *resource allocation/management* in data centres [3]. The key concept of SDN is to separate the control plane from the data plane and concentrate the core logic on a *software-based controller* [4]. A controller can be deployed to handle operations within a single domain however, there will be need for inter controller communication (east-west interface) to maintain ***reliability*** in forwarding IoT traffic to gateways. The SDN control plane can either be organised as flat or hierarchical structure. In the hierarchical plane, a master controller provides synchronisation between the lower-level administrative controller domains. On the other hand, controllers may communicate directly with each other when the flat structure is utilised.

The integration of FEC and SDN [3] has immense prospect for speeding up practical deployments and federating resources for IoT. However, there are avenues that still need to be investigated to bridge the gap between FEC and SDN. First, *scalability and Byzantine fault tolerance* of SDN control plane is understudied for wider or large-scale geographic coverage. Since a distributed form of SDN control plane is required to communicate with adjacent controllers, further investigation into scalability and robustness of the SDN plane is therefore essential. Second, previous research only focussed on the virtualisation and management of SDN controllers in a wired network. To serve a mostly mobile community in the future, the benefits of SDN and current standards such as OpenFlow should be utilised by wireless networks [5] for federating nodes.

As human beings, we naturally have a "built-in" sense of fairness for allocation of scarce resources in FEC data centres – essentially, this sense of "fairness" is commonly referred to as max-min fair allocation policy. In many real-life application instances, there are situations where equal division of resources among IoT offloading users is not the desired plan [6]. Existing research on fair allocation for distributed SDN has focused more on centralised e.g., Dominant resource fairness [6] and distributed or Nash bargaining-based (stochastic) policy. However, the advent of fog and edge computing (FEC) with *sandbox execution environments* (containers) has increased the need for more *scalable and robust* mechanisms to support priority scheduling for joint multi-resource and multitasking offloading in distributed data centre environments (wireless networks inclusive) comprising administrative controllers at the SDN control plane.

### 1.1.1 Fog and Edge Computing (FEC)

By computing in proximity, the latency problem can be solved for offloading in IoT. The on-device processing approach should ensure that data sent over the network could be handled immediately. This is essential for latency-sensitive applications, such as autonomous vehicles, where having to wait milliseconds may be untenable [7]. The distributed nature of FEC also mitigates *the bandwidth problem;* processing data would start at the point of collection and only stored data is pushed to the cloud, making edge computing more efficient and scalable. Reduced network load and the distance data must travel for edge application services ensures lower latency and minimises the cost of transmission.

**Edge Computing.** Edge computing, as depicted in Figure 1-1, takes advantage of closeness to the physical items of interest by exploiting the relationships those devices may have with one another. Data coming from the physical world of IoT or *edge devices* via sensors is processed and then used to change the physical environment state via actuators. As analytics and knowledge generation happens at the edge, the communication bandwidth between the controlled systems and centralised data centre is much reduced. **Mobile edge computing (MEC)** considered a key enabler of next-generation networks (e.g., 5G) comprises base stations and access points equipped with edge computing servers to handle application requests at the edge of the network. Edge computing is essentially an extension of technologies such as remote cloud services, self-healing networks and peer-to-peer networking. As compared to traditional architectures, edge computing offers many advantages such as optimising resource usage in a cloud computing system. *Offloading* at the edge of the network mitigates network traffic which in turn reduces the risk of data bottlenecks. In edge computing, embedded computational capabilities of local resources or MEC datacentres are accessed through ad-hoc networking and used to process IoT data from task offloading.

**Fog Computing.** Fog computing offers infrastructure and software services through distributed fog nodes to execute IoT applications within the network. Fog enables ***computation offloading by*** moving the workload closer to the network edge thereby reducing latency because of data travel and bandwidth. Computation offloading in fog computing happens across one or more distributed nodes in the network. Traditional networking devices like routers, proxy-servers, micro-data centres can act as fog nodes and create a wide area cloud-like services either in an independent or *hierarchical clustered approach*. In the hierarchical fog network architecture, data from endpoints is transmitted through gateways, processed, and return transmission.

From a ***broader perspective,*** edge computing is naturally considered to be a subset of fog computing. Essentially, it is impractical to deploy and manage fog without integrating edge computing technologies. Conceptually, the difference in the two concepts lies in the location of intelligence and placement of computing power. With ***Fog computing***, compute power and intelligence are situated at the *local area network* whereas the concept of edge computing sees intelligence and compute power placed at ***devices***.

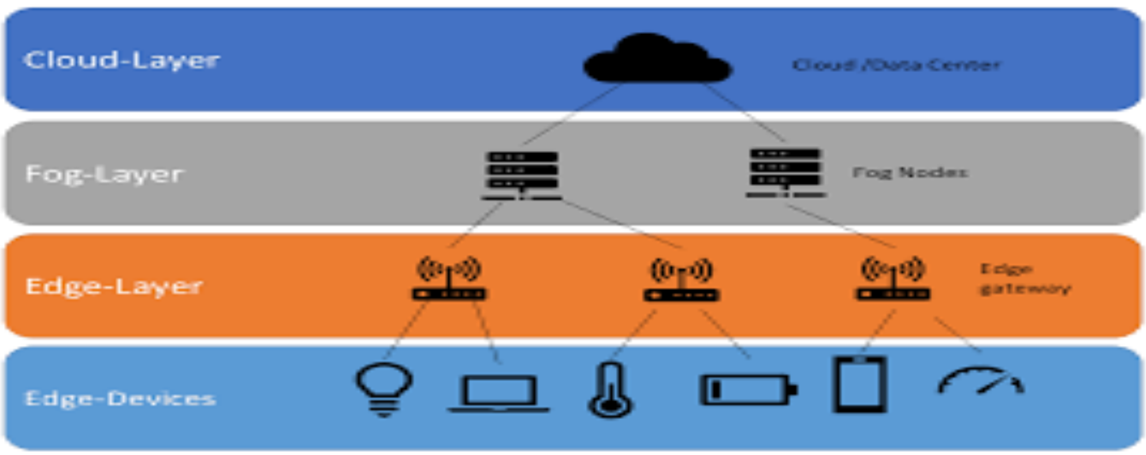



Figure 1-1. Fog and Edge Computing (FEC) Paradigm

### 1.1.2 Load balancing and Quality of service (QoS)

Network management and orchestration in Software defined clouds (**SDC**) recognises the problem of joint provisioning of *administrative hosts and network resources as well as scheduling*. Two conditions must be satisfied in joint host and network resource management/orchestration for computation offloading: (1) finding the minimum subset of hosts and network resources that satisfy a given workload and (2) meeting SLA and user's QoS requirements e.g., latency. The problem becomes more challenging when SDC supports virtual network functions (VNF) and service function chain (SFC).

*Load balancing* for performance isolation in FEC is generally regarded as a critical part of resource allocation in the implementation of ***SDN technology***. Opportunistic scheduling and reservation of virtual network resources is difficult to achieve because of many edge devices and dynamics. In addition, *SDN scheduling mechanisms* should be designed to assign ***shared resources*** in a non-manipulative manner. One of the most critical issues for distributed IoT applications is *priority allocation of all network resources*. Once allocation of multiple resources is not managed properly then heterogenous IoT services may experience reduced allocations – applications would not receive a fair share of resources. Fair allocation policies are made worse due to the dynamic nature of IoT user applications in distributed data centre environments leading to FEC scheduler manipulation.

The International Telecommunication Unit (ITU) [8] defined the concept of QoS as the degree of conformance in delivering service to the user by the provider based on agreement between them [8]. Many IoT applications need huge processing speed and large data storage to enable delay-sensitive control, and high-speed broadband networks to stream video or audio. Clouds provide a back-end solution for dealing with large data streams (offloading) - processing them for the unprecedented count of IoT devices and humans in real-time [9]. The sheer number of different interactions and links between IoT objects qualifies it to be a scalable complex system. Furthermore, some IoT services are required to be reconfigurable and composable at run-time for Quality of Service (QoS) guarantees. The difficulty in satisfying dynamic requirements of IoT services [10] [11] becomes more evident. In IoT systems, more dynamic QoS attributes are concerned, such as ***accuracy***, the required ***network resources***, and the ***coverage of IoT***. To solve the difficulties mentioned above, new QoS models for service oriented IoT is required [11]. For joint computation offloading, the resources should be allocated in such a way to be at dynamic equilibrium - load balancing for performance isolation is meant to increase system performance as stipulated in service level agreements (SLAs). *Load balancing* on different virtual networks and ***QoS*** can degrade significantly unless *efficient monitoring and control* is imposed [3].

## 1.2 Thesis Scope and research questions

A multi-controller design can effectively improve the performance of SDN in comparison with a single controller. Different types and physical locations of controllers may suffer from attacks and failure (indeterminate) which in turn influences the ***reliability*** of the control plane. An unbalanced distribution of controller workloads will degrade the network performance, and how to load balance multi-controller task workloads is also an important point of multi-controller SDN research - placing multi-controllers requires an effective approach to cope with the challenge of ***complexity.***

### 1.2.1 Thesis problem description

Software defined networking (SDN) controllers [3], also called orchestration platforms perform the role of network packet “brokers” between client IoT applications (northbound interfaces) and network processing elements (southbound interfaces). SDN controllers are logically designed as centralised entities generally used to support loosely coupled IoT applications (offloading). However, FEC being a distributed computing paradigm, suggests distributed monitoring and feedback can provide an efficient solution for load balancing and QoS-awareness.

In a multi-controller architecture for resource allocation, an intelligent software controller agent because of limited resource capacity or capabilities of nodes may not accomplish many tasks in offloading. In these situations, agents can strategically cooperate by forming groups to solve the problem [12]. Forming coalitions or clusters that are effective is a major research challenge in the field of multi-agent systems (MAS). Logically distributed node clusters can be viewed as groups of cooperating physical FEC nodes, working together on given tasks, short-lived and goal oriented. The coming together of distinct autonomous controller nodes to behave as a coherent unit (called a resource cluster) is called coalition formation; It has long been studied in cooperative game theory (GT) and has emerged an important application in MAS. A clustering mechanism typically involves three computationally challenging processes namely: (i) Coalition value calculation (ii) Coalition structure generation (iii) Reward calculation [13]. As singleton **FEC nodes** may not complete a task individually, ***software agents*** would therefore exchange information and try to form node clusters which give them the best efficiency.

The concept of complex adaptive systems is a prominent feature of the complexity theory paradigm. Complex adaptive systems absorb information from the environment and create knowledge content that helps decision-making [14]. The system complexity is derived from the diversity of, and the level of interaction between its constituent parts; arising from the collective control that the parts exert on the overall system.

Complex adaptive systems are used to describe "nonlinear systems" whose behavior is established by the interaction of its adaptive parts; this explains how learning and innovation occurs in living systems [15], [16]. The parts are diverse in their ability and form [17]. Consequently, the larger the number of constituent parts, the higher the level of interaction between them, making it difficult to predict the behaviour of the system [18].

Collectively these parts cause the system to sway in a certain direction, which may be hard to predict because each part of a complex system acts according to ***its own best interest***. The parts would constantly be seeking to improve value thereby pushing the system away from equilibrium [17]. Over time, the extensive interaction between the parts would determine the behaviour of the overall system within its environment. Essentially, the constituting parts should learn from interactions and reorganise themselves to better adapt to the environment [19].

### 1.2.2 Fair allocation characteristics of multi-resources

In many heterogeneous and distributed scenarios for FEC, and to provide high-quality (fault tolerant and scalable) offloading services to end-users [3], the distributed SDN control plane would need to guarantee the required fair characteristics – namely load balancing or sharing incentive, strategy proof, envy-freeness, and Pareto-optimality.

**Sharing incentive (Load Balancing):** FEC user applications are better off sharing the cluster than using their individual partition. A FEC user should not be able to allocate more tasks in a cluster partition consisting of 1/n of all resources – for a cluster comprising similar nodes and n user applications [6].

**Strategy proof:** FEC user applications (offloading) should not be able to manipulate the scheduler by being untruthful about their resource demands. This attribute is responsible for incentive compatibility, a FEC user application cannot improve allocated resource by misreporting requirements.

**Envy-freeness:** This attribute embodies the notion of fairness - A FEC user should not be envious (or prefer) of the allocation of another user.

**Pareto optimality:** It should not be possible to increase the allocation of a FEC offloading user without decreasing the allocation of at least another user. This property is important as it leads to maximising system resource utilisation subject to satisfying the other properties.

### 1.2.3 Fair allocation issues for distributed SDN Control plane

*Ghodsi et al* [6] commented on the strategy-proofness (or truthfulness) and load balancing for performance isolation or sharing incentive properties, which they suggested are of special importance in data centre environments like FEC.

Generic:

- ***Group strategy-proofness*** is important: "Anecdotal evidence from cloud operators that we have talked with indicates that as it is common for FEC nodes to attempt to manipulate (knowingly or unknowing) schedulers" [6].
- An allocation policy that satisfies the ***sharing incentive*** attribute also guarantees ***load balancing*** [20] for performance isolation. It also ensures a minimum allocation to each

offloading application. i.e., a FEC user tasks cannot do worse than owning a fair share (i.e., 1/n) of the cluster resources – regardless of the demands of the other FEC users.

Specific (Network topology):

The flat and physically distributed SDN control plane [21] for the federated edge must partition multiple ***tasks*** or workloads from service requesters, called users or clients between the providers of resources or services, called edge servers [22]. Host applications require multiple resources of the edge server, shared between clients. However, it is uncertain whether the application of marginal integrative value for the resource clusters under consideration will shift the entire platform closer to the Pareto optimum, unless optimum conditions are met in the rest of the economic system [23], [24]. Generally, the latter will not be the case, and as such, first-best Pareto optimality may not be efficient in the case of resource allocation mechanisms (RAMs) seeking to control dynamics (externalities) in controller bargain [20] strategies for joint computation offloading. In such instances, second-best Pareto optimality [25] may be more efficient.

**Ageing or faulty SDN control plane**. Monotonic (population and resource) scheduling to address load balancing and QoS challenges in joint computation offloading is an important problem for the federated edge. This is because of "the ageing process" or network effects (same-side and cross-side) that arise due to cooperative and competitive controller bargaining strategies at the flat SDN control plane.

- **Population monotonicity:** When a FEC data centre agent relinquishes host resources, none of the allocations of the remaining user applications (offloading) should decrease.
- **Resource monotonicity:** If more non-excludable resources are added to the federated edge, the allocations for existing user applications (offloading) should not decrease.

For the hierarchical and physically distributed SDN control plane [21] [26]**,** fog network [27] layers are divided into at least three (3) discrete layers – client, distribution and cloud layers. Each tier or layer is responsible for carrying out ***multitask*** processing at the edge of the network. The distribution layer would always connect to the cloud through gateways.

**Ageing or faulty SDN control plane.** Because the distribution layer is an aggregation point for the devices in the client application layer, gateway, and low-level controller's placement optimisation (also called bottleneck fairness) is an important problem for fault-tolerance.

- **bottleneck fairness**. If there is one fog network resource (i.e., gateway) that is percentwise demanded most of the time by every user application, then the problem should reduce to max-min fairness for the gateway's resources.

### 1.2.4 Research questions

An important challenge for network designers is the number of SDN controllers and their positions in distributed network architectures will impact the overall performance of the control plane. The distributed control of SDNs using multiple controllers face a series of pressing design and performance challenges that require special consideration [28]. These issues include (1) Complexity/scalability (2) Fault tolerance/ reliability (3) adaptive consistency (4) Interoperability, (5) Monitoring and (6) Security.

**Controller Placement Problem (CPP).** Heller et al. [29] proposed to control three factors to determine how many controllers to use and where to place them. First of them is the required response limits, specifically latency agreements. Second, resource availability, fairness of state of the network, processing, and bandwidth, and finally, the network topology. The research questions in this thesis are limited to two important yet unaddressed fair resource allocation problems related to network topology of distributed software defined networking (SDN) control plane for joint multi-resource and multitask applications in FEC.

**Research question I: How can monotonic and throughput scheduling be achieved for the joint multi-resource and multitask offloading considering an ageing (faulty) flat SDN control plane of the federated edge comprising administrative data centres or edge-clouds (ECs)?**

**Monotonic Scheduler.** In flat design, the network provides more resilience to failures as controllers can communicate directly with each other on the same layer. However, the task of managing autonomous controllers becomes harder. At the federated (platform) edge, controller-to-controller bargain externalities give rise to opportunities for Pareto improvement. In this research, we address population and resource monotonicity for joint multi-resource and multitask offloading via a truthful scheduling mechanism to achieve allocations that are Second-best Pareto optimal (SBPO) [30] which accounts for controller-to-controller bargain externalities in the resource allocation mechanism.

**Research question II: How can bottleneck fairness and cross-layer throughput scheduling be achieved for joint multi-resource and multitask forwarding/offloading considering an ageing or faulty (wireless) hierarchical SDN control plane?**

**Bottleneck fair (link) Scheduler**. The hierarchical design gives a simpler way to manage the physically distributed controllers, however, because of the upper layer of the control plane, the bottleneck problem or problem of a single point of failure exists. In a hierarchical controller architecture, it is common to have about three layers. Each layer contains a type of controllers. Typically, the bottom layer contains the low-level or local controllers, while the upper layer contains one gateway or root controller, which means that we have the problem of a single point of failure (bottleneck), even if it concerns only one layer of the control plane.

For maximum throughput in gateway and low-level controller's placement, the wireless hierarchical network should be strategically split into distributed clusters covering all IoT mesh nodes in the network graph, each cluster has a gateway or cluster head. The bottleneck fair scheduling policy optimisation is treated as an NP-complete problem. We propose a truthful scheduling mechanism to address this huge optimisation problem by breaking it into subproblems and then solve them recursively.

## 1.3 Aim and objectives

**Thesis Aim**. The aim of this thesis is to develop truthful scheduling mechanisms for rejuvenation of decentralised controller deployment in Software defined networking (SDN) technology to address priority offloading in FEC.

**Overall thesis objectives.** Essential design and performance objectives for two studies are outlined as follows:

**Design Objectives:**

- **Centralised SDN controller architecture:** Mechanism and multi-controller design based on physically distributed SDN controller nodes or administrative data centres in ***flat and hierarchical control planes.***
- **Group strategy proofness and Sharing incentive (load balancing):** For this objective, scheduler manipulation and load balancing for performance isolation are important in the design of fair allocation policies for joint offloading in FEC. In the case of FEC or distributed environments, it is necessary to ensure that allocation policies are group strategy proof or truthful. Load balancing (sharing incentive) guarantees node assignment to controllers in problematic cases to avoid congestion and workload imbalance between controllers – providing performance isolation.

**Performance Objectives:**

- **Computational complexity:** the time-taken for the run-time scheduling scheme should be fast.
- **Resource utilisation**: To evaluate the effect of load balancing on resource utilisation optimisation, we propose the utilisation ratio (bandwidth) to measure the SDN controller node resources corresponding to the IoT workloads or links.
- **Quality of service**: To measure the QoS in distributed SDN architectures, numerous metrics are used. The most significant for load balancing in FEC environments are described below:
    - ✓ Costs: Quality costs or costs of quality aims to quantify the total cost of quality related efforts and deficiencies.
    - ✓ Latency: This metric is the time spent by the physical network switch to transmit the incoming data traffic. it is subject to the network's congestion status and the

load of the forwarded tasks. The time taken for a packet to be transmitted from the source to the destination is called transmission latency.

- ✓ Response time: It is defined as the interval of time between request acquisition, receipt, and response. The metric indicates the SDN's ability to meet QoS requirements of applications in terms of resource availability, delay, and reactivity.
- ✓ Throughput: By definition, throughput is the quantity of information that is processed and forwarded from source to its destination during a specified period in other words, the processing speed of the SDN controller nodes and its performance is referred to as throughput typically computed as throughput = 1/ response time. Successful load balancing schemes that assign IoT workloads to SDN controller nodes with capacity sufficiency would maximise throughput.

## 1.4 Research methodology

Applied research [31] is mainly concerned with finding a solution to an immediate problem facing a society, or business organisation. This thesis adopts applied research methodology to address fair allocation problems for rejuvenation of multi-controller SDN in intelligent FEC environments.

**Reverse game theory** provides the *mechanism design* [32] via the study of mathematical models to analyse the resource allocation strategies of distributed *software controller agents* to make decision, which will result in social welfare amongst them.

**Software cybernetics and homeostasis.** The study of both cybernetics and homeostasis provides a transdisciplinary approach used to explore regulatory systems that focuses on how systems apply information, models, and control actions to navigate towards and maintain their goals.

With the help of reverse game theory, *this thesis* applies cybernetic and homeostasis principles of *positive or/and negative (distributed)* feedback to develop and analyse hybrid scheduling schemes for joint multi-resource and multitask offloading in fault-probable multi-controller SDN for FEC. The proposed schemes are implemented as multi-agent reinforcement learning (MARL) algorithms to provide deterministic allocation policy. We realised the hybrid (and truthful) schemes as logically centralised scheduling policy control mechanism applying physically distributed node offloading/forwarding strategy feedback for rejuvenation of the flat (federated edge) and hierarchical (wireless fog-mesh network) control plane.

Although multi-agent reinforcement learning (MARL) and game theory (GT) have different origins, they share a common approach on how to define the resource allocation problem in the game environment, that is, states, agents, and policies (strategy profiles). In critical evaluation of the proposed fair scheduling policies, resource allocation games are formulated and solved as multi-agent markov decision process.

To carry out experiments that evaluate the novel resource allocation mechanisms, it is necessary to extend some classes in CloudSim [33] to represent more precisely the federated edge, only few classes need to be modified for iFogSim - the wireless fog environment. The results from several runs of controlled simulated experiments using the CloudSim [33] and iFogSim [34] is then analysed using MATLAB.

## 1.5 Original contributions

The main (generic) contributions to literature as summarised as (**see Chapters 4, 5 and 6**):

- Scalable and robust (fault-proof) control learning frameworks for max-min fair scheduling of joint multi-resource and multitasks. The framework applies software cybernetics and homeostasis to ensure monotonicity (population and resource) and bottleneck fairness in the distributed SDN control plane.
- A mechanism (and controller) design of hybrid scheduling schemes for monotonicity and bottleneck fairness as logically centralised scheduling policy control applying physically distributed node offloading strategy feedback related to homeostasis.

- We propose to model the reference architecture of the federated edge as Inter-edge coalition games (ICGs) with controller-to-controller distributive bargain externalities. Whereas the proactive reference architecture of the wireless hierarchical fog is modelled as constrained coalition (CCF) games.
- We apply multi-agent reinforcement learning (MARL) with robustness against adversarial controllers to enable efficient joint multi-resource and multitask offloading/forwarding enabling the rejuvenation of decentralised controller deployment.
- We propose both IP-DRF and Dfog-DRF schemes implement joint policy search via generalisations of control algorithms and DRF (and DRFH) scheduling. The MARL schemes apply multi-objective (lexicographic) optimisation to guarantee efficient joint policy for monotonicity (population or resource) and bottleneck fairness in multi-controller SDN.
- Empirical results via simulation (CloudSim & iFogSim) to evaluate the efficacy of both IP-DRF and Dfog-DRF scheme for rejuvenation of faulty distributed SDN control plane applying the generalised max-min scheduling methods for deterministic joint policy.

## 1.6 Thesis structure

The rest of the chapters in this thesis is organised as follows:

**Chapter 2** discusses application areas for software cybernetics research - networks systems (Software define networking), IoT and autonomous cloud computing. We review published works for resource management of SDC, decentralised control methodology and gaps in literature related to fairness-aware offloading at the federated edge and wireless hierarchical fog networks are analysed.

**Chapter 3** discusses the methodology used in the design of hybrid scheduling schemes for priority-based joint offloading in distributed SDN control planes. The research methodology and justification – alternative methods and proposed methods (Robust MARL against adversaries), theoretical framework, optimisation model and approach – approximate and exact (anytime) optimisation.

**Chapter 4** presents the design of the rejuvenation of decentralised controller deployment applying (policy-based) MARL for monotonic scheduling at the SDN-based federated edge. The work in this chapter answers *research question I* of this thesis and details contributions to literature.

**Chapter 5** addresses *research question II* by proposing a fog co-forwarding scheme that applies distributed (game-theory based) clustering methodology for priority scheduling. The research study addresses truthful scheduling with bottleneck fairness in wireless hierarchical fog networks via throughput maximisation of fog gateway and low-level controller's placement.

**Chapter 6** discusses experimental evaluation of fair allocation policy for rejuvenation of decentralised controller deployment at the federated edge and Wireless fog environments based on the Java-based CloudSim architecture. This chapter first focusses on the Cloud market model along with extensions to the existing CloudSim framework. Furthermore, it discusses the experimental setup and results obtained from several simulation runs.

**Chapter 7** provides the summary, revisits the research questions, significance as well as limitations and future work.

# Chapter 2 Literature Review

- SDC enabling technologies: service-oriented computing, network systems (Software defined networking), IoT and computation offloading, autonomous cloud computing.
- Resource management policies for joint host and network systems.
- Centralised optimisation and decentralised (independent) learning for multi-controller management**:** priority scheduling for joint offloading at the: (1) flat SDN control plane of the federated edge and (2) hierarchical SDN control plane of fog-based wireless ad-hoc networks is discussed.
- The concept of software cybernetics for mechanism and controller design/ management in distributed SDN control plane - theoretical and Artificial intelligence-based methods.
- The gaps in reviewed literature are analysed.

## 2.1 SDC technologies for Offloading in FEC

To enhance resource management in Data centres, Software-Defined Cloud Computing (SDCC) utilises the advances in areas of cloud computing, virtualisation, SDN and NFV. We briefly introduce in this section for FEC, several concepts and technologies that enable SDC [35].

### 2.1.1 Service Oriented Computing

Service-oriented computing (SOC) is a computing paradigm that utilises services as basic resource for developing applications. In the next phase of distributed computing, they will be vital for development because services issue a uniform and standard information paradigm for a wide range of computing devices. Developers may put together existing web service components to create new applications for complex service requirements.

Applications in Service-based Systems can often be viewed as the composition of various computing services following specific workflows. Techniques based on Service Oriented Architecture (SOA) to enable utility computing to have emerged and become a cost-effective way for organisations to outsource their computing tasks to infrastructure providers and receive computing services on-demand. Software cybernetics approach is proposed by [36] to modelling and solving timing and resource constraints to deploy and schedule workflows.

Liu et al. [37] proposed a control-based approach to the security adaptation problem in adaptive service-based systems. A performance index that incorporates security requirements and delay deadlines is proposed to transform the problem into an optimisation problem. To demonstrate the feasibility of the approach and experimental data showing that the system provides desirable balance between security and delay requirements, the proposed security technique is implemented in an example application.

Whereby each component fulfils specified functionalities, Cloud applications are typically composed of multiple cloud service components communicating with one other through web service interfaces. One of major bottlenecks for enhancing availability and efficiency of complex and ageing cloud application systems is lack of effective fault tolerance scheme. Liu et al. [38] proposed an adaptive failure detection and ageing degree evaluation approach to predict which cloud service components deserve foremost to be rejuvenated and a component rejuvenation approach based on checkpoints with trace replay is proposed to guarantee the continuous running of cloud application systems.

### 2.1.2 Software Defined Networking (SDN)

**SDN** is one very promising solution that decouples the data and control planes, incorporating one centralised controller for the network. This increases the flexibility to control and manage the network as desired, thereby creating many new possibilities [39]. Adami, et al. [39] developed a system to enable QoS control and routing in Software Defined Networks. Whenever the OpenFlow controller installs a rule for a flow, it also places it in the right queue. The experimental results show a more efficient way of managing the network resources and providing guarantees about handling traffic. To manage adaptation behaviour of complex network systems, Ravindran [40] applied software cybernetics to assess the quality of adaptation in a network system in the presence of uncontrollable external environment conditions.

**Multi-Controller SDN**. In a *flat architecture,* the controllers are placed horizontally on one single layer or level. More specifically, the control plane consists of just one layer, and each controller has a partial view of its network and the same responsibilities at the same time [21].

SDN controllers can also be positioned vertically in a *hierarchical or vertical architecture*. The controllers are allocated among multiple levels, which therefore means that the control plane has several layers, generally two or three [21]. The controllers have different responsibilities, and they can take decisions based on a partial view of the network.

Santos et al. [41] proposed Decentralize-SDN or D-SDN, a framework enabling not only physical but also logical distribution of the Software-Defined Networking (SDN) control plane.

The adaptive resource capacity manager, the policy configuration manager, and ***physically distributed monitoring interfaces*** [42] make up the proposed intelligent management layer for SDN. The basic function of automatic capacity manager at any given time is provide optimal decision about the best or most appropriate *logically centralised policy* to execute at the core of the management layer [43]. The responsibility of the configuration manager is to control dynamic behavior for multiple SDN controllers [44].

Monitoring in the SDN-based environment [45] should be continuous to facilitate decisions as part of overall resource utilisation optimisation. Most importantly, monitoring can be carried out ***passively or actively***. For passive monitoring, one or more entities collect relevant information. The entity may continuously send polling messages to nodes asking for information or perform this on demand [46]. On the other hand, monitoring is active when nodes are autonomous and decide when to send asynchronously state information to the central entity.

Centrally placed between the application and control planes in the SDN architecture [43], the architecture has a IoT *traffic router,* capable of forwarding big data packets in real-time. The router is designed to support fast processing of IoT application streams. The concept of *IoT traffic gateways* is the basic abstraction provided by the IoT router situated at the edge of the network. Conceptually speaking, SDN controllers act to direct IoT data streams to one or more nodes in network processing elements through *dedicated gateways* specifically set up to mirror the *economics* of shared resources at the edge of the network. There is still a lack of technical know-how on the actual deployment of multiple controllers using control software.

### 2.1.3 Internet of Things and Computation Offloading

The Internet of Things (IoT) refers to a wide set of interconnected and intertwined devices and things that provide substantial value to stakeholders. The IoT is being actively developed with the emergence of Smart Cities, populated with large amounts of smart objects that are used to deliver a host of services. The architecture of IoT is dependent on the pervasive presence of smart objects or “things”, which addresses several new challenges in the software engineering domain. These include orchestrating large-scale smart objects, service discovery, data gathering, and data processing and so on. To enable resource discovery, Perera & Vasilakos [47] suggested how IoT resources could be described using semantics. To develop this, their research proposed a knowledge driven approach referred to as Context-aware Sensor configuration model (CASCOM) to simplify the configuration of IoT middleware platforms.

Computation offloading is regarded as a technique that enables an IoT device or low powered device to forward the processing of a task. This task for computation offloading could be code, service to a higher capabilities and resources. Essentially, a *typical offloading system* consists of two components, a mobile client or device and a server at the cloud or edge. However, large-scale offloading like in Fog and edge computing does not allow efficient distribution and utilisation of multi-resources in the centralised cloud system [48].

### 2.1.4 Autonomic Cloud Computing

Zhu (2012) in a keynote speech at COMPASAC'12 addressed many issues of cloud computing and software cybernetics, such as service architecture, agent-based computing, metrics and software evolution [49]. With the recent growth of IoT based applications, the use of cloud services is increasing exponentially. The next generation of cloud computing must be sustainable to fulfil the end-user requirements which are changing dynamically. Dynamic and appropriate resource allocation is a crucial problem in cloud computing.

Clouds have become the dominant computing environment of the current and the next decade by delivering all kinds of services, focusing on innovative applications, large-scale resource sharing, and high-performance orientation. The key aspects of cloud computing have been reported in the definition provided by the National Institute of Standard and Technologies (NIST): "Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction" [50]. Software engineering issues need to be addressed to make the most effective use of the clouds.

With Clouds being complex, large-scale, and heterogeneous distributed systems, resource management is a challenging task. Resource management requires automated and integrated intelligent strategies for provisioning of resources to offer services that are secure, reliable, and cost-efficient. As such, effectively managing services becomes mandatory in software platforms that constitute the inner fabric of computing Clouds. Autonomic computing provides a path towards controlling cloud computing services [51]. Open issues in autonomic resource provisioning and present innovative management techniques for supporting SaaS applications hosted on Clouds was discussed by Buyya, et al. [52]. Their work presented a conceptual architecture and early results evidencing the benefits of autonomic management of Clouds. [53] discussed one of the case studies of the ASCENS project; a vision of an autonomic cloud: A cloud which is based on voluntary computing and using peer-to-peer technology to provide platform-as-a-service. It applied self-awareness and self-adaptation as the main components for the execution management of arbitrary applications. Despite this, many aspects of this technology require further attention, such as large-scale tests, alternative implementation models, self-adaptation performance in the cloud etc.

An important feature of cloud computing is elasticity, and this can be understood as how a computational cloud fits variation in their workload by provisioning and de-provisioning resources. Autonomic Computing brings many concepts such as control loops and thresholds-based rules which are important in the construction of elastic cloud computing solutions. Coutinho, et al. [54] proposed an elastic architecture for cloud computing based on concepts of Autonomic Computing. Konstanteli et al. [55] proposed a mechanism using probabilistic optimisation model, for admission control of a set of horizontally scalable services. The proposed model reduces the resources required to assure a given quality of service by employing statistical knowledge of the elastic workload requirements of services.

## 2.2 SDC Resource management policies

A core functionality required for any manufactured system is resource management. Managing resources affect the three basic criteria for evaluating computer systems, which include cost, functionality, and performance. Essentially, inefficient resource management has a negative peripheral effect on functionality and central impact on system performance and cost.

There are different strategies for SDC resource management associated with the three cloud delivery models, Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). However, in all delivery models, the cloud services providers are faced with enormous, vacillating loads that challenge the claim of cloud elasticity. In some cases, when a spike can be predicted, resources provisioning can be planned or takes place in advance. For example, seasonal web services may be subject to spikes.

Typically, SDC resource management requires complex policies and decisions for multi-objective optimisation. It is extremely challenging because of the complexity of the system, which makes it extremely difficult to obtain accurate global state information. Resource management is also subject to incessant and sometimes unpredictable interactions with the environment. The importance of ascertaining the value or performance of different public cloud providers has engendered the development of monitoring services (see Section 2.3.2) that report metrics to provide a better picture of the real behaviour of the different services.

***Admission Control***. The objective of an admission control policy in cloud computing is to prevent the system from accepting workloads that would lead to the violation of high-level system policies. For instance, the system may likely reject an additional workload that would prevent it from completing work already in execution or contracted for action. To limit the workload in a dynamic system requires some current knowledge of the global system state. A mechanism using probabilistic optimisation model was developed by [55] for admission control of a set of horizontally scalable services. The presented admission control test finds the optimum allocation using the optimisation model, which incorporates business rules in terms of the trust, eco-efficiency, and cost, and considers affinity rules of the components that comprise the service. The model reduces the resources required to assure a given quality of service by employing statistical knowledge of the elastic workload requirements of services.

Service level agreement (SLA) is signed between service providers and clients with the intention to provide the best suitable performance for consumers. The Quality of service (QoS) is that aspect of resource management that is the most difficult to address and, at the same time, possibly the most critical to the future of cloud computing. Interestingly, strategies used for resource management often jointly target performance and power consumption.

***Capacity allocation***. Capacity allocation means allocating cloud resources for individual instances. Here, an instance refers to a service activation. Locating resources that are subject to multiple global optimisation constraints requires anyone to a search a large space when the state of individual systems is changing so rapidly. Ardagna, et al. [56] developed a capacity allocation algorithm to ensure SLA and handle fluctuating workloads. The allocation algorithms interact with geographically dispersed resource controllers and can redirect the load whenever congestion is present in the network. Moreover, it requires that an application is run on multiple virtual machines or VMs and workload is evenly distributed on the VMs. As the workloads vary, a workload predictor is used to forecast future workload requirements and resource capacity is changed based on resultant load forecasts.

In cloud computing, a critical objective is reducing the cost of providing the service, interestingly, this also means minimising energy consumption. This gives a different interpretation of the term load balancing. There is a correlation between load balancing and energy optimisation, and they affect the cost of providing services. The key concept is that instead of having the load evenly distributed among all servers, it is possible to concentrate it and use the smallest number of servers while others are switched to standby mode, a state of the server which uses less energy. Al Sallami [57] proposed an artificial neural network (ANN) based load balancing technique. This technique uses back propagation method to distribute the load equally among all the servers. Each user's demand is predicted, and resources are allocated according to the predicted demand, but active servers at any given time depend on the demand of users at a specified time. Consequently, active servers are minimised leading to low energy consumption. The paper also highlights the relationship between energy consumption and carbon emissions. Research in [58] predict workload and used a central controller to allocate (and manage) resources dynamically to the running applications. In their effort, resource management decisions are carried out hourly to minimise the overhead incurred because of the decisions. The resource management decisions include power up, shutdown, and VM migration from one server to another and consume a considerable amount of energy and network resources.

Motivated initially by the need to save power for mobile devices, dynamic voltage, and frequency scaling (DVFS) techniques such as Intel SpeedStep and American microdevices (AMD) PowerNow lower the voltage and the frequency to decrease power consumption. These techniques have migrated to virtually all processors, including those used in high-performance servers. Because of their lower voltages and frequencies, the processor performance decreases. However, this comes at a substantially slower rate than the energy consumption. Ali et al. [59] proposed an idea to maximise the revenue of service providers by meeting SLA and minimising energy consumption. To achieve their aim, a framework has been proposed in which servers have a dynamic voltage/frequency scaling (DVFS) module. However, they assumed that a server could not be switched on or off and there is no common cost of VM migration. Furthermore, they proposed a hybrid optimisation technique to solve issues related to load balancing, DVFS, resource allocation, and service placement based on VMs.

In early research efforts, much attention has been given to optimisation solutions (cost/energy) focussing only on either the host or network ant not both. Many research approaches that do attempt to address joint host and network resource provisioning either suffer from high complexity or faulty behaviour. Therefore, it is essential to design scalable, load balanced and fault-proof scheduling policies for joint offloading at the distributed SDN control plane considering both *administrative hosts and network resources* at the same time. This is necessary for FEC because optimisation of one can exacerbate the situation for the other. Necessarily, there are about four mechanisms or enabling techniques for the implementation of resource management policies in cloud systems [60], these are:

- **Control theory:** predicts only local behaviour, control theory applies feedback to guarantee system stability and predict transient behaviour.

- **Machine learning (ML):** One major advantage of applying machine-learning techniques is the absence of a performance model of the system. This technique is applied to coordinating several autonomic system managers.
- **Market-oriented/economic mechanisms:** The methods would not necessarily need a system model, as is the case in combining auctions for bundles of resources.
- **Utility-based:** Utility-based methods use a performance model and a mechanism to correlate user-level performance with cost.

## 2.3 Federating (Inter-)edge resources

The federated edge network issues need an optimisation approach that can simultaneously handle: (1) an enormous number of services with replications (2) devices and computing nodes joining or leaving the network and (3) heterogeneity, separate internal operations and characteristics, different service providers [3].

Resource allocation approaches in the federated edge marketplace should address networking issues via: (1) loosely coupled and geographically dispersed allocation of resources (2) provision of isolated execution environment for multiple services and applications (3) user mobility, real-time awareness and (4) the avoidance of over subscription on individual clouds.

### 2.3.1 Architectures: Resource provisioning

In cloud systems, resource provisioning directly concerns the decision about the number, types, and location of resources to be deployed for a specific application. Grozev and Buyya [61] proposed taxonomies for Inter-cloud architectures and presented detailed surveys of each project. Inter-clouds are broadly classified in architecture as *volunteer federation*, in which a group of cloud providers voluntarily collaborate to exchange resources, and Independent Inter-cloud, where multiple clouds are used by an application or its broker. Volunteer federations can be further classified as *centralised or Peer-to-peer*. In a centralised architecture, there is a central entity that either performs or fosters resource allocation. For example, if a provider cannot satisfy the request to provision resources from its client, the request is then redirected via the centralised entity to another provider that can offer the needed resources. Whereas, in the peer-to-peer architecture, cloud providers communicate and negotiate directly with each other without the need for a centralised server.

On the other hand, Independent Intercloud approaches enable resource provisioning from multiple clouds without direct exchange between providers as in the previous approach [62]. This is achieved with an independent service or library that supports multiple cloud providers. These approaches provide application provisioning, hosted either externally or in-house, and libraries, which enables a uniform approach to facilitate the use of multiple clouds.

### 2.3.2 Projects: Platforms, monitoring and selection

The ***Intercloud*** [63] project developed at the University of Melbourne, Australia, is one of the first initiatives for Interconnected cloud computing. Its proposed architecture is centralised and built around a central entity called the *Cloud Exchange* (CEx). The *Cloud Exchange* (CEx) plays the role of a market maker in bringing together service producers and consumers. It aggregates the infrastructure demands from the application brokers and evaluates them against the available supply currently published by the *Cloud Coordinators*. The *Cloud Coordinators* provide programming, management, and deployment environment for applications in a cloud federation.

Jrad et al. [64] proposed a generic architecture for a Cloud service broker operating in an Inter-edge environment using Cloud standards. The goal of the broker is to find the most suitable Cloud provider while satisfying the users' service requirements in terms of functional and non-functional Service Level Agreement parameters. Their effort discussed the broker value-added services, as well as an elaborated broker design. They focused especially on the incorporation of expected SLA management and resource interoperability functionalities in the broker. The proposed architecture was validated and evaluated using a realistic simulation testbed.

The Resources and Services Virtualization Without Barriers (***RESERVOIR***) project is a European initiative that extends previous research on interconnected grids [65]. The architecture of ***RESERVOIR*** is peer-to-peer and hence does not rely on a central entity. In the

***RESERVOIR*** model for inter-edge or federated edge resources, inter-edge comprises two or more independent cloud computing providers. The brokerage of multiple clouds is achieved via *Claudia*, an abstract layer for execution of services on top of a cloud federation. The scaling process is automated through a mechanism (elasticity rules) used to specify the dynamic capacity requirements of an application at deployment time.

The ***mOSAIC*** project addresses the problem of interoperability by defining a common ontology and it aimed at developing an open-source platform that enables applications to negotiate Cloud services as requested by users [66]. The ***mOSAIC*** open-source API and platform allow for the development and deployment of applications spanning multiple clouds. Using a Cloud ontology, applications can specify their service requirements and communicate them to the platform via the innovative API. The platform implements a multi-agent brokering mechanism that searches for services matching the applications' request and could possibly compose the requested service if no bespoke service is found**.** Right Scale provides a consolidated Dashboard and APIs to manage many clouds. They provide a configuration framework with templates to set up the VMs easily. In addition, they provide an easy management tool on multiple cloud providers, but do not perform the provider selection

Some *InterCloud* libraries have also been developed in recent years, Apache Jclouds5, for example, is a Java library for Java-based interaction with various providers. This provides a provider-independent API for execution of operations regarding provisioning of storage and computing resources. Many of these Inter-edge libraries allow application developers program their own application-specific broker.

*CloudHarmony* reports for a wide set of public cloud providers, the results of benchmarks about performance, network, and uptime [67]. Monitoring services located inside and outside of the cloud provider collect these metrics. In addition, and on behalf of *CloudHarmony,* these services execute some benchmark applications. A Monitoring tool called Cloud Provider View is provided by *CloudSleuth* and displays the perceived response time and the percentage of availability of various providers in different locations. The tool works by continuously monitoring the performance and availability of a test application deployed in each data centre

[68]. They provide the status of each data centre by analysing the collected data from applications running in each data centre. This can bolster comparison between different data centres and providers.

CloudStatus is a monitoring tool that collects, in real-time, observations of infrastructure metrics such as availability, response time, latency, and throughput from Amazon and Google cloud services. These metrics are aggregated by the server from sources inside and outside of the provider and calculated with the aim of diagnosing the health of the cloud. Applying the diagnosis outcome, they provide an overall status of the cloud in real-time that can affect the performance of the applications running in the cloud. The results cover global availability and normalised metrics across the cloud instead of monitoring specific instances of the cloud.

Garg et al. [69] proposed a paucity in software framework to automatically index cloud providers based on their needs and as such, proposed a framework and a mechanism to measure the quality and prioritise Cloud services. The framework aims to influence significantly and create healthy competition among Cloud providers to satisfy their Service Level Agreement (SLA) and improve their Quality of Services (QoS). SMICloud [69] is the framework to rank cloud providers for a given application considering the service measure indexes (SMI). The SMI comprises accountability, agility, assurance of services, cost, performance, security and privacy, and usability. It works by assigning various Key Performance Indicators (KPI) to evaluate the indices in different cloud providers.

Rak et al. [70] developed a cost/performance evaluation technique using benchmarks and simulation to evaluate the trade-off between costs and performance of cloud applications. More specifically, given a mOSAIC cloud application, it is possible to predict performance indexes and resource consumption under generic workloads. This supports deployment on the resources of the provider that guarantees the desired performance levels and minimises the costs for executing the application. The authors proposed the use of non-functional requirements to create a system that suggests the best option from a set of cloud providers.

### 2.3.3 Resource management in Inter-edge: Enabling techniques

Aoyama & Sakai [71] proposed that implementation of resource management for Intercloud must include a unified way of managing various resource configurations for each service such as servers, storage units, and networks. They suggest the aim of this function is to manage resource configurations necessary to prevent services from degradation or disaster. Most research study already carried out have trivially addressed brokerage in the Intercloud environment, and so far, no complete resource management model has been presented [72].

Takefusa et al. [73] proposed Grid ARS, an advanced reservation-based resource management framework, which aims to integrate heterogeneous resources and construct a performance-assured virtual infrastructure over Federated clouds environment. *Grid ARS* was developed using common web services technologies and standards and provides four services that address resource management, resource allocation planning, provisioning, and monitoring of the constructed virtual infrastructure.

Cloud broker is an important feature of Intercloud computing, which plays its role in terms of resource management, service discovery, service-level agreement negotiation and match-making between service providers and customer(s). Aazam and Huh [72] proposed a service-oriented dynamic resource management model, which covers cloud service consumer characteristics, and includes the issues of resource prediction, customer type-based resource estimation, and reservation, advanced reservation, pricing, refunding and acquired the quality of service-based refunding. The system was implemented and validated using *CloudSim 3.0.3* (The CLOUDS Lab, The University of Melbourne, Australia) toolkit. Their method was also evaluated on Google cluster trace comprising 12,000 machines.

Son, et al. [74] proposed an architecture to address the challenges emerging from the perspective of the system administrator. Using the architecture, administrators acquire the desired number of VMs from the best provider with proper resource size that covers their non-functional requirements. Their proposed approach would assist system administrators to setup the required IT infrastructure and migrate their applications to the cloud without much concern of computing the required amount of resources. The method translates the high-level non-functional requirements from administrators into VM resource parameters, selects an adequate type of VM and the provider, and allocates actual VMs from the selected provider. The prototype shows, based on non-functional requirements, that the system is effective in receiving requests and provisioning resources on different cloud providers.

A distributed control (algorithmic based) approach was developed by [75] for performance management of services hosted in distributed cloud broker environments. The proposed distributed algorithm adopts the interaction balanced based approach, where with respect to system dynamics, services are decoupled from each other, however, the services are coupled in terms of overall deployment wide operating cost functions. These Services are coupled by a limited amount of system resources in a cloud broker infrastructure. The focus of this approach is to maximise profit for both the service provider and broker while ensuring optimal resource allocation. Their research effort introduced a novel negotiation approach between the broker and the various cloud providers for optimised allocation of resources through interactive bidding in cloud computing environment.

Research effort in [76] and [77] developed a substantial body of work in agent-based cloud computing that provides empirical evidence to show that multi-agent systems are appropriate software tools for automating complex interactions within an Intercloud. Sim [78] modeled economic encounters between clouds in A federation as a coalition game. His research addressed two major issues in Inter-edge interaction. First, the process through which clouds agents choose its coalition and secondly, the division of Intercloud coalition payoff among the data centre agents. Within a Federated cloud, each self-interested EC controller negotiates and establishes agreements with other controllers to meet its own objectives and to optimise its own payoff.

For clouds to discover and select their coalition members and to fairly divide the payoff of the Inter-edge coalition, Sim [78] devised a novel four-stage cloud-to-cloud interaction protocol that governs how data centre agents join coalitions, and a strategy profile of the data centre agents that converges to a sub-game perfect equilibrium. His work provides a set of mathematical theorems and proof to show that: (i) Intercloud coalition formation strategies converge to a subgame perfect equilibrium and (ii) every Data centre agent in an Intercloud coalition receives a payoff equal to its shapely value.

### 2.3.4 Federated edge resources and allocation

For the main study, we reviewed related work in Inter-edge scheduling and allocation for joint offloading – the networking problem, sharing, aggregation and bargaining techniques in distributed cloud markets - the flat control plane, centralised and decentralised allocation policies in offloading for IoT.

#### 2.3.4.1 Network management problem

The Eaas platform [79] and ENORM [80] framework address resource discovery, deployment, and ***load balancing*** in the context of individual edge nodes. However, they do not assume federated edge resources.

Addressing the resource allocation challenge of the federated edge with service-centric, multiple administrative clouds, and reliability issues requires a solution with characteristics of both a centralised control and distributed model [3].

A. Agent-Based Cloud Workflow (Flow scheduling)

Adopting an agent paradigm [76] enables fog and clouds to maintain autonomy and interact more intelligently and efficiently via social interactions [78], [81]. It allows agent-based fog and Inter-edge resource allocation systems to be designed with desirable properties specified and proven using GT [62], [78], [82], [83].

Cloud Workflow (Scheduling) using Petri-nets. Whereas Gutiérrez and Sim [77] developed a multi-cloud (with brokers or intermediaries) model in which agents manage workflow-using coordination by synchronization, Frincu et al [84] developed an agent-based multi-cloud scheduling model using event condition and action rules. In this model, agents are implemented as a monitor-analyse-plan-execute (MAPE) control loop whereby each agent has four modules:

monitor, negotiator, scheduler, and executor. Agents manage the workflow by 1) considering the set of actions to be performed to produce a set of plans and 2) selecting from the plan's actions to execute. We propose a centralised (agent-based) workflow model in our work, Inter-edge (i.e., DRF-H [85]) scheduling policies for execution of task offloads are implemented at host and VM level based on time (space)-sharing.

B. Federated edge schemes – Capacity sharing, aggregation, and Bargaining

**Cloud federation.** Celesti et al. [86] developed an agent-based testbed comprising home clouds (HCs) and foreign clouds (FCs), whereas HCs are clouds requiring additional resource capacities from other clouds, FCs lease a portion of their resource capacity to HCs. In their work, clouds establish federations to enhance their own resource utilisation and to enlarge albeit temporarily, their resource capacities. The process of establishing a cloud federation is helped by discovery agents (DAs), matchmaking agents (MAs) and authentication agents (AAs). In our work, cloud federation (sharing incentive) can be established via either integrative or distributive negotiation to enhance resource utilisation and increase their resource capacities.

**Hedonic game**. Mashayekhy et al. [87], [88] designed a cloud federation mechanism, which allows the cloud providers to make their own decisions to form a federation yielding the highest total profit. In the proposed mechanism, federations of cloud providers decide to merge and split to form a federation providing requested resources as a service to the user. The policy-based mechanism also determines the individual profit of each participating cloud provider in the federation. Each cloud provider covers its incurred costs and obtains a profit based on its market power.

**Bargaining mechanisms**. Mathematical proofs in [89] validate that the agent-based bargaining mechanism in [81] enables vertically arranged fog and cloud resources to be optimally priced. A list of research issues is involved when building Inter-edge infrastructure including operability, interaction, and economics [90], [52].

**Parallel negotiation in Multiple Cloud Markets**. Sim [82] developed a multilateral negotiation mechanism comprising consumer, broker and provider agents called the bargaining position estimation (BPE) for making concessions. The BPE strategy is a market-driven in the sense that an agent adjusts its concessions in response to different market situations [83].

#### 2.3.4.2 Fairness and throughput-aware computation offloading

Centralised control refers to the use of a single inter-domain SDN controller that makes joint scheduling decisions for computation offloading at the federated edge. On the contrary, when decision-making for joint offloading is distributed across the edge nodes or controllers, we refer to the scheme as distributed or fully decentralised [91].

##### A. Centralised controller optimisation

Multi-objective integer programming is commonly used for generating deployment and redeployment allocation policies for scheduling workloads in grids, clusters, and clouds [92]. Mixed integer programming (MIP) models implement asset fairness whereby allocations are truthful and Pareto-optimal but violate the sharing incentive (***or load balancing***) and are impractical for much larger problems [93] without some means for dimensionality reduction. Dominant resource fairness [6] (DRF) and its variant [85] for heterogenous edge servers is a generalisation of the Max-min fairness for multiple resource types to satisfy important properties however, the centralised scheduling approach (weighted DRF) is not easily scalable to distributed cluster environments like the federated edge.

##### B. Decentralised controller (multi-agent) learning

***Nash Equilibrium and Load Balancing***. Grosu and Chronopoulos [94] formulated the *load balancing* problem in heterogenous distributed systems as a non-cooperative game among users. In their proposed non-cooperative game for load balancing, the structure of the Nash equilibrium (NE) policy is presented. Their work also suggests performance achieved in a noncooperative game at the Nash Equilibrium may be made better in a cooperative game setting (Nash bargaining solution - NBS).

***Reinforcement learning to achieve Nash Equilibrium***. Zheng et al [95] proposed the mobile users' offloading decision process under dynamic environment as a stochastic game. Their work prove that this is equivalent to a weighted potential game, which has at least one Nash equilibrium. Chen [96] designed a decentralised computation offloading mechanism that can achieve a Nash equilibrium of the game and quantify its efficiency ratio over the centralised optimal solution.

Research work in [97] formulated the energy efficient edge-server activation problem in a MEC offloading system using minority games. The distributed decision-making mechanism focused on minimising server energy while guaranteeing QoS. Using auction theory, a resource allocation mechanism [98] was proposed whereby service providers in the mobile edge network design contracts with the edge node infrastructure providers. Contracts enable service providers to efficiently provision assigned computational tasks and schedule offloaded tasks in a way that latency is minimised. Considering a multi-cell, quasi-static environment [99], the computation offloading problem is cast as a dynamic sequential game. The work establishes the existence of Nash equilibrium and develops a distributed convergent offloading scheme.

Sim [78] devised a novel four-stage cloud-to-cloud interaction protocol that governs how data centre agents (will) join coalitions, and a federated edge policy (strategy profile) of integrative EC strategies that converges to a sub-game perfect Nash equilibrium, and results in each agent obtaining a fair share of the total profit that is equal to the Shapley value.

Applying the CEEI (Nash bargaining) policy to solve the fair allocation problem for joint offloading at the federated edge may not yield ***group strategy-proof or fault-tolerant*** solutions particularly when node offloading strategy feedback related to sparse interactions (population and resource non-monotonicity) from distributive EC bargaining is ignored.

## 2.4 Network management for Wireless hierarchical fog networks

In the second study, we review fog computing and its optimisation taxonomy, interference-aware controller placement problem in ***wireless hierarchical*** fog networks, particularly methods that incorporate fairness. The review also identifies current gaps in cluster-based methods and highlight the contribution of our game-theoretic approach for the dynamic and distributed link scheduling problem in hierarchical WMN.

### 2.4.1 Optimisation Taxonomy: Fog Architecture and dynamic life cycle

Fog computing allows for computation offloading of IoT applications from central clouds all through to the network edge. From end users' devices (mobiles) to many in-network facilities along the cloud-to-things continuum, computational tasks are offloaded [100], [101]. To ensure quality of service (QoS) requirements for latency-sensitive applications, fog networks use the computational capabilities (resources) of edge network devices with proximity to end users [102]. Optimisation problems in fog network systems can be grouped depending on which tier(s) of the three-layer model (Cloud, edge resources and end devices) is involved. Real fog computing involves at least two of these layers. In addition, the different stages of the service life cycle also influence the optimisation approach. This service-life cycle dimension is further classified into design-time, deployment-time, and run-time optimisation. For fog networks, optimisation for reconfiguration needs to happen at run-time. Sardellitti et al [103] considered the joint computation offloading problem in mobile edge computing whereby some end devices offload compute tasks to a nearby edge resource. For each task of the end device, it is decided, whether the jobs in the case of offloading, which radio communication channel should be used. Considering several end devices with a potential for channel interference, the complex (nonconvex) problem is solved by means of heuristics.

### 2.4.2 Gateway (controller) placement in Wireless fog (-mesh) networks

Gateway placement in wireless networks as well as clustering and coverage problems in wireless ad-hoc networks have been under study [104], [105], [106] regardless of the underlying physical topology and protocols used (WSN, WMN, M2M or IoT).

**Minimising gateways**. The gateway placement problem in [107] aimed at minimising the number of gateways, while ensuring access points bandwidth requirements. Their effort considers the issue as an instance of the network flow problem, allowing multipath routing. However, the proposed greedy heuristics leads to non-optimal solutions when constraints on communication path length are imposed. Moreover, since gateways are usually placed whenever the rest are fully served, the iterative greedy approach unbalances the load at the gateways.

**Network Performance.** In Hierarchical WMNs, proper placement of gateways is a key factor in the optimal throughput, load balancing on the gateways and satisfying QoS requirements [108]. Gateway placement in areas with low traffic or few numbers of Mesh clients (MCs), means they might be underutilised. It is for this reason most research has been dedicated to the gateway placement problem to enhance the network performance as compared to minimising the number of gateways [109].

### 2.4.3 Hierarchical fog clustering

Grouping of a set of IoT application nodes into classes of similar (fair) computing nodes is referred to as ***distributed clustering***. ***Load balanced*** gateway placement in Wireless hierarchical fog networks is essential to SDN performance [91].

Zeng et al [110] address the load balancing gateway placement problem and proposed a greedy algorithm GA-LBC to partition WMN into load-balance and disjointed clusters, each cluster satisfies QoS requirements. Their work proposes a hybrid algorithm HA-LBPG based on GA-LBC algorithm and the genetic algorithm to get the near-optimal solution.

Significant delay occurs at each hop in a Hierarchical mesh network (multi-hop) for offloading due to contention for the wireless channel, packets processing, and queueing delay [108]. Essentially, the delay constraint is considered a function of the number of hops between the sensor and the IoT gateway. The total IoT traffic in each cluster is bounded by the capacity (size) of the gateway, based on its connectivity and processing capacity and speed. The bottleneck on throughput is reduced to the load of congested intermediate wireless links. Since traffic is aggregated and forwarded by intermediate wireless access points (WAPs), we refer to the load on individual wireless links as relay load in unit of IoT traffic.

Using a spanning tree rooted at each cluster head (i.e., gateway) for message delivery, Bejerano et al. [107] adopted a clustered view of the WMN. Breaking the distributed problem into two subproblems and solving each one separately, their work proposed polynomial time approximation algorithms solutions within a constant factor of the optimal ones.

### 2.4.4 Throughput-aware task forwarding schemes

We review fair link scheduling techniques for the interference-***aware*** (strategy proof) load balanced ***gateway (and low-level controller's) placement*** problem that consider wireless QoS requirements.

**A. Heuristic-based optimisation**

Benyamina et al [111], ***without the use of constraints*** for optimisation, developed the cluster-based gateway placement algorithm (CBGPA) to address end-to-end bounded delay communications and attempt to handle scalability using a nature inspired meta-heuristic optimisation model.

- To optimise placement of gateways in wireless networks, some other studies [112], [113], [114] have applied ***heuristic methods*** such as genetic algorithm, simulated annealing, and tabu-search meta-heuristics. The mobility of wireless fog network client devices regarding gateway placement in dynamic WMNs is also studied in more recent research [115] [116]. A swarm optimisation method that harnesses the social relationship notion of users, in which groups move with high probability to the same direction when they share the same interests [115]. These solutions exhibit very fast optimisation speeds but may not be fair because they easily violate solution accuracy – metaheuristics trade-off solution accuracy for convergence speed.

**B. Centralised controller optimisation**

Liyanage et al. [117] introduced a hierarchical distributed controller architecture for software defined vehicular network (SDVN) with the top tier of controllers regionally distributed on the Internet and the bottom tier of controllers are placed in several selected roadside units (RSUs) closer to the vehicles to control the latency of the system. They present a novel controller placement model for RSU level controllers and based it on the facility location problem. The location of RSU and delay are optimised (heuristically) as an integer quadratic programming problem.

Wenjia et al [118] proposed a novel gateway placement approach to address the problem, in which three primary objectives are optimised, i.e., the number of gateways, the average MR (mesh router)-GW (gateway) hop count and the variance of gateway load. The problem is first modeled as a multi-objective linear programming problem, followed by a ***two-stage load balancing*** gateway placement algorithm. The first stage is a weight-based greedy gateway selection, and the second stage is **load balanced** MR attachment.

Researchers in [119] also studied the optimal placement of a given number of gateways on a wireless fog backbone network to achieve maximum throughput. The problem is formulated as an ILP and a greedy algorithm that defines the gateways' locations to optimise the cross-layer throughput is given. Due to interference in case of simultaneous transmissions, the capacity reduction on wireless links is probed. The algorithm aims to provide truthful or interference free link scheduling. However, a distinction between gateway and client devices is not made and the installation cost is predefined, since the number of gateways is given a priori.

**Asset fairness** violates ***sharing incentive (load balancing) and bottleneck fairness*** as well as suffers from the curse of dimensionality with an increase in problem size; moreover, the greedy heuristics approach provides fast short-term solutions but provides no guarantees on optimality (sub-optimal) for the fog scheduling problem.

### C. Decentralised controller (multi-agent) learning

***Reinforcement learning to achieve Nash equilibrium***. Because radio communication channel is usually shared in wireless networks, the behavior of a wireless device may affect the communication capabilities of other neighbouring devices. Nash bargaining solution (distributed control) is applied to communication networking to solve routing and resource allocation problems [120] in distributed cluster environments [121], [122].

Cardellini et al [123] proposed a three-tier architecture, which considers a usage scenario where no central authority exists, and multiple non-cooperative mobile users share the limited computing resources of a close-by cloudlet and can selfishly decide to send their computations to any of the three tiers.

**Independent Learners.** Lin et al. [124] proposed QoS-aware adaptive routing (QAR) via reinforcement learning in multi-layer hierarchical SDNs via a three-level design of controllers i.e., super, domain and slave. Their work first introduced a distributed hierarchical control plane to minimise the signalling delay, serving as a realistic SDN architecture. Furthermore, QAR was then proposed to enable adaptive, time-efficient, and QoS-aware packet forwarding upon the hierarchical architecture.

In [125], an offloading mechanism based on the Stackelberg game is proposed for cloud-enhanced vehicular networks with edge computing capabilities. Servers and offloading vehicles are modelled as leaders and followers. Proof of the existence of Nash equilibrium is provided and a distributed algorithm is designed to maximise the edge server’s utility while satisfying latency constraints of tasks.

## 2.5 Decentralised controller (multi-agent) learning for resource allocation in FEC

The intelligent Software defined networking (SDN) controller not only monitors the status of the networking nodes and links but can also be incorporated with service monitor functionality using a northbound application. As such, it becomes much easier to incorporate a customisable scheduling policy mechanism that can guarantee fault-proofness and ***performance isolation*** for applications considering joint ***host (compute) and network resources.***

Decentralised controller deployment is considered as a more scalable option when compared to the centralised controller deployment. However, because FEC execution environment is inherently decentralised, multi-agent reinforcement learning policies are typically trained in a centralised manner. The evolution of SDN and OpenFlow allows for a logically centralised but physically distributed control plane**.** Data traffic may be forwarded through at least two administrative domains – belonging to different service providers. There is a need for abstraction and control over disjoint administrative domains using multiple controllers. As the **FEC control plane** can be structured/organised (physically) as either ***flat or hierarchical***, there is also a need for interdomain and inter-controller traffic communication that maintains reliability in forwarding traffic to the gateways. Indeed, for the distributed SDN, multi-controller deployment is envisioned to address the ***scalability issue*** raised by the centralised architecture of SDNs.

**Game theory** (GT) is concerned with strategic interaction among several decision-makers, where each player knows that his actions affect the well-being of the other players, just is their actions affect his. In GT, the term “game” is used to describe interactive encounters between several rational participants. Each game has predetermined rules of engagement. These rules e.g., joint venture agreements define the structure of the game. The motivating forces are the actions or strategies that participants can employ. It is also assumed that each participant can rate the outcome of the encounter, which depends on the actions of all the participants, and based on their own order of priorities. These outcomes are expressed in numeric form and are called the payoffs or levels of utility of the various available strategies' players can take [126]. GT is conventionally divided into cooperative and non-cooperative games; each having its own solution concepts, e.g., for non-cooperative games, minimax value and Nash equilibrium, core, and Shapley value for cooperative games. However, elements of both competition and cooperation are present in most real-world economic and political interactions.

**Multi-agent reinforcement learning – GT perspective**. Resource allocation or stochastic games are identified by a set of available resources for each player in which the players can access each other through a communication network. The access cost among the players is determined by communication network, and the collective goal is to satisfy customer needs at minimum cost.

The theories of bargaining [127] and coalitional form games [128] use the same 'welfarist' assumption that only utility possibility sets are relevant to the analysis. It is assumed that these sets summarise the opportunities available to groups of agents from exploiting unspecified resources under their control. To identify systematic methods of selecting one or several utility profiles, for each configuration of these feasible sets is the objective of the analysis. It uses several axioms based on the shape and relative positions of these sets or involving comparisons across configurations. Rahwan [129] focussed on the coalition structure generation (CSG) problem for cooperative and competitive game settings. The CSG problem [129] proposed partitioning a set of agents into a structure to maximise social welfare.

The coalition structure generation (CSG) problem poses the question, how can we divide the number of autonomic elements into groups to maximise performance? The CSG problem is somewhat challenging computationally it belongs to the group of problems classified as NP-hard. Therefore, to solve this problem efficiently, even under restrictive assumptions, researchers are required to develop a range of algorithms applying approximate or exact techniques.

**Pareto-Optimality**. Considering both competitive and their cooperative aspects, the seminal work of Nash [127] pioneered the notion of a solution concept for strategic games. Nash [127] defined such a solution for two-person games and proved the existence and uniqueness theorem. The solution is derived by means of "bargaining with variable threats". In the first competitive stage, each agent declares a threat strategy, to be used if negotiations break down; the outcome resulting from deployment of these strategies constitutes a disagreement point. In the next cooperative stage, agents aim to achieve *a Pareto optimal structure* and share the gains relative to the disagreement point; the controller agents coordinate their strategies, and the sharing of resources is done using *principles of proportional fairness*. From a computer science perspective, Nash equilibrium-based solution is not robust [130] or *global optimality* is not guaranteed.

While in normal form games the challenges for reinforcement learners originate mainly from the interactions between the controller agents, in Markov games they face the additional challenge of environment state transitions. This means that the controllers typically need to combine equilibrium solvers from repeated games with MDP approaches from single-agent RL.

**Nash-Q learning.** Nash equilibrium as a solution concept for learning in resource allocation gives rise to a few criticisms. The most basic of them is that Nash equilibria need not be unique, which leads to an equilibrium selection or stability problem [131]. In other words, multiple Nash equilibria can exist for a resource allocation game. Furthermore, these equilibria may also differ in the value attributed to the controllers. This further suggests that a method learning Nash equilibria, cannot guarantee a deterministic outcome or more basic, a unique payoff for the controllers.

Zhu and Lu [132] proposed the robust implementation of *decentralised controller learning*, the distributed Dominant Resource Fairness (D-DRF) scheduling policy is based on a fitness heuristic and a task forwarding mechanism. Although the hybrid scheduling scheme improves performance at some scale using distributed localised resource managers, however, it overlooks an implementation of *node offloading strategy feedback* to learn interdependencies among controllers from joint action at the SDN control plane. We are motivated by the concepts of software cybernetics and homeostasis to address *group strategy proofness (or fault-tolerance) as well as problem complexity* for joint multi-resource and multitask offloading in FEC.

**Joint action Learners**. Most literature on the CSG has traditionally been approached as a *single objective function* for hybrid scheduling policy control. Moreover, the global utility function of learning agents in a coalition structure [133] [134] can be generalised to include the concept of weights for *Pareto optimisation* using lexicographic ordering. Figure 2-1 shows a taxonomy of fair allocation policies for joint offloading that applies to FEC and forms a basis for the discussion in the sub sections of this chapter.

## 2.6 Software cybernetics

Yang, et al. [135] reviews new trend and further classified software cybernetics as (i) software cybernetics I based on the first-order cybernetics. First order software cybernetics is typified by feedback loop control e.g. modeling software systems using finite state machines, and (ii) software cybernetics II is based on the higher order cybernetics; characterised by system developers, software under development and execution environments influencing each other to form a more complex system.

Achieving *scalability* while maintaining *reliability* requires a relevant ***control distribution scheme*** that considers both the organisation of the SDN control plane and the physical placement of SDN controllers [28].

Software cybernetics is motivated by whether and how software (SDN in this case) behaviour can be controlled. The underlying motivations and ideas for software cybernetics include feedback mechanisms in software process. Logical foundation for control systems, as well as software fault-tolerance and complexity have already been identified as potential research areas [136].

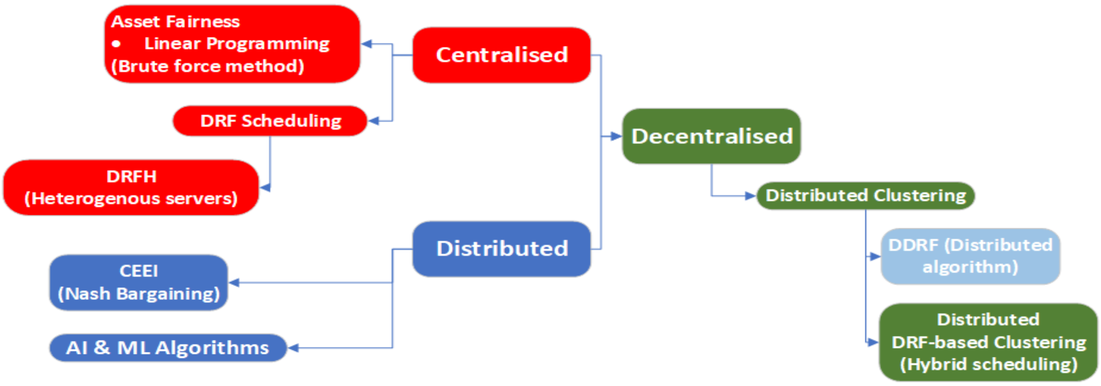



Figure 2-1: Taxonomy of fair offloading policies for the distributed SDN control pane (FEC)

### 2.6.1 Traditional models

Different mathematical methods are applied to design effective system models, which constitutes the main methodological approach in software cybernetic research. Dynamic system models, formal models like the extended finite state automata and controlled Markov chain exemplify model-based approaches. Supervisory-control theory is based on the finite state automata used to represent discrete-event dynamic systems.

First-order cybernetics or software cybernetics I [135], steers the system into the stable state via negative feedback processes, which is designed to take place in isolation from its environment as a closed system.

In prior research work, linear dynamic system models were developed to describe software service behaviours and the software test process. Cai [136] viewed software testing as a control problem and devised a control Markov chains (CMC) approach to determine optimal test strategy. The CMC approach provides theoretic justification that for some circumstances a Markov model matches the software test profile. Hu, et al. [137] proposed a new adaptive software testing approach based on the improved CMC, which aimed to replace several presumptions adopted by previous models with more realistic situations in software testing.

The finite state machine (FSM) is a classic example of a formal model in software cybernetics. Gaudin & Bagnato [138] described a set of safe behaviours as FSMs and these FSMs represent over-approximations of the behaviours of the system to be controlled, relying on a sound Supervisory Control Theory. The extended finite state machine (EFSM) is used to model communication software behaviours [139]. Using extended finite state machine (EFSM) as an embedded part of the controlled system, Yang & Gohari [140] have presented a framework to implement supervisory control map. The research work also shows that the constructed EFSM shows the same behaviour as the supervised system. Wang and Cai, et al [141], developed algorithms that transform EFSM for specification and description language (SDL) to the control model of discrete event systems (DES). Their research work shows that EFSM is expressed as **a closed loop control system.** Lorenzoli, et al. [142] proposed the GK-tail algorithm; a technique that uses interaction traces to automatically generate EFSM models of the behaviour of software systems.

Zhao et al. [143] provided efforts aimed at improving the GK-tail algorithm and they propose an improved method for modelling software behaviour based on EFSM. Their work designed and implemented a software behaviour modelling system to verify the efficacy of their improved method. Wang & Cai [141] investigated the supervisory control problem of restrict EFSM model and proposed a necessary and sufficient condition and an optimal algorithm to the supervisor. The promising result related the software design problem to supervisory control theory and enriches the research content of software cybernetics.

### 2.6.2 Artificial Intelligence

Advances in the field of Artificial Intelligence has fostered active research in software cybernetics. Specifically, software engineering has become an important application area for the application of machine learning techniques. Fuzzy logic is a typical instance of the logic-centric or rule-based approach used in software cybernetics research; it is a knowledge-based formal model for machine learning. Addressing challenges or uncertainty in complex software systems, Yang, et al. [144] applied fuzzy based logic to control complex software systems. The goal of this fuzzy-based technique was to develop a self-adaptive executable software framework that improved the performance of process control mission-critical systems. Ding et al. [145] proposed an adaptive control system based on fuzzy logic and update the controller itself with a set of fuzzy rules. Yau et al. [146] applied the principles of software cybernetics in service-based systems (SBS) to synthesise controllers for online adaptation and monitoring. This approach also included situation-aware planning for offline resources taking as input timing and resource constraints. Park & Yeom [147] used the concept of feedback in software cybernetics to propose an approach for validating Semantic web rule language (SWRL) rules. Their method comprises preparation, structural analysis, contextual analysis and the SWRL rule adaptation. The approach constituted a feedback loop whereby the SWRL rule to be validated is the controlled object while the validation of SWRL rules is the controller. With third-order cybernetics, many researchers see the introduction of artificial intelligence taking software cybernetics research to a new level [135].

## 2.7 Gaps

**Research gap I.** *Joint-offloading schemes* [97], [95] in the flat and distributed SDN control plane apply multi-agent learning (MAL) to achieve Nash equilibrium (or CEEI) from micro-economic theory guaranteeing ***load balancing for performance isolation*** [20]. However, allocations from learning for Nash equilibrium may violate group strategy proof as well as **population and resource monotonicity**. Essentially, FEC controller nodes may manipulate the SDN scheduler about their resource requirements and reduce the allocation of other FEC applications. In addition to violating group strategy proofness, when resources are added (non-excludable ones) or relinquished (node failure) by FEC server nodes, user applications may experience reduced allocations or degradation thereby violating ***monotonicity.***

On the other hand, asset fair allocation policies do not guarantee *load balancing for performance isolation* and resource monotonicity characteristics. Applying the asset fair policy, it is possible for FEC client applications to suffer performance isolation issues. In addition, resource non-monotonicity may happen i.e., reducing resources allocated to some FEC applications while increasing others. Dominant resource fairness DRF [6] and DRFH [85] policy satisfies load balancing, group strategy proof and Pareto optimality but violates resource monotonicity. Moreover, DRF is centralised and not easily scalable (throughput) due to physically distributed nature of the flat SDN control plane.

**Research gap II.** Nash bargaining solution, the multi-agent learning methodology from CEEI theory guarantees ***bottleneck fairness*** for joint forwarding/offloading in wireless hierarchical networks. However, CEEI-based policies may manipulate the hierarchical fog scheduler **–** *it is not fault-proof*. On the other hand, a more straightforward approach using *asset fairness* ensures strategy proofness but would violate sharing incentive or load balancing for performance isolation and bottleneck fairness. *DRF allocation policy* guarantees group strategy proofness (or truthfulness) and bottleneck fairness however, it is centralised and would not scale efficiently (single gateway) when applied to the wireless hierarchical (and distributed) SDN control plane.

**Summary**. We propose the concepts of *software cybernetics and homeostasis* to fill gaps in literature of SDC resource management policies for joint offloading in FEC, particularly the *scalability and byzantine fault-tolerance* issues [28] of the centralised and distributed (RL-based) schemes respectively. We aim to propose rejuvenation of decentralised controller placement via the application of multi-agent reinforcement learning (MARL) with robustness against adversarial controllers. Our proposed MARL scheme for resource allocation would comprise *logically centralised scheduling policy control* applying *physically distributed node offloading strategy feedback* to provide dimensionality reduction – decrease and conquer as well as divide and conquer. Cybernetic and homeostasis principles (positive or/and negative feedback) in reverse games are used to develop efficient resource allocation mechanisms [148] for the federated edge and wireless hierarchical fog networks considering multi-resource and multitask (distributed) offloading.

## 2.8 Summary

Software cybernetics provides insights into software engineering problems of emergent behaviour in service-oriented architecture, self-adaptive architectures, the role of software metrics in control and evolution of FEC systems.

- ✓ With the widespread use of distributed computing and cloud computing in our daily life, the ubiquitous role of software systems suggests that for software cybernetics to add significant value to modern software systems.
- ✓ Research gap I: In addition to monotonic scheduling, the fair allocation mechanism also needs to provide group strategy proofness, second-best Pareto optimality, as well as problem dimension reduction. We consider non-monotonicity in FEC from positive or negative Controller-to-controller bargain externalities as multi-controller offloading strategy feedback to adapt (rejuvenate) the faulty (ageing) flat SDN control plane.
- ✓ Research gap II: To mitigate fog scheduler manipulation, the problem of fog gateway and low-level controller's placement (joint) policy optimisation or bottleneck fairness for priority-based joint forwarding is concerned with QoS constraint dimensionality reduction. We consider cybernetic and homeostasis principles of positive and negative multi-controller forwarding strategy feedback to proactively rejuvenate (adapt) the faulty distributed control plane of wireless hierarchical fog networks.

# Chapter 3 Research Design and Methods

- Research methodology: Multi-objective problem and solution model, Markov or stochastic games, value iteration and joint policy for MARL.
- Software cybernetics and homeostasis principles of positive or/and negative feedback are proposed to design truthful resource allocation mechanisms (RAMs) for joint multi-resource and multitask offloading/forwarding in FEC.
- Theoretical methods: hybrid scheduling mechanism and controller design for joint offloading in distributed data centre environments based on Partition form game (PFG) and constrained coalition formation (CCF) game.
- Critical appraisal: Large state-action spaces and incomplete information.
- Solution accuracy and complexity trade-off: exact versus non-exact methods.

## 3.1 Research aim

The goal of this research is to develop hybrid scheduling policy mechanisms for joint multi-resource and multitasking offloading in FEC that satisfy essential fair allocation characteristics (see Sections 1.2.2 & 1.2.3) including (1) monotonicity (population and resource) and (2) bottleneck fairness.

## 3.2 Scheduling games for joint offloading

If the SDN environment controllers experience is Markovian and the controller agent is allowed to try out sufficient actions for task offloading, some RL algorithms (e.g., Q-learning) are certain to converge to the optimal strategy. However, markov decision process (MDP) do not offer the same theoretical support for multi-agent learning (MAL) even though a solid mathematical framework for single-agent learning is well-established. The reward a controller agent receives may depend on the actions taken by other controllers when multiple adaptive controllers interact with each other, thereby rendering the Markov property invalid since the SDN environment is no longer stationary and each controller is therefore faced with the classic moving-target problem. Markov or stochastic games and joint action learners (JAL) are extensions of the MDP framework.

### 3.2.1 Markov decision problem formulation

Stochastic games generalise strategic-form games to dynamic situations in Fog and Edge computing environment and MDP to multiple decision makers as response to player actions. The solution to the markov decision problem is directly concerned with multi-agent reinforcement learning. The reinforcement learning problem can formally be described using the framework of stochastic (Markov) games. A stochastic game is a tuple $(n, S, A_1, \ldots, A_n, T, r_1, \ldots, r_n)$

Where,

$$S \text{ is the set of environment states}$$

$$A_i \text{ is the set of actions available to agent } i$$
$$r_i: S \times A \to \mathbb{R} \text{ is the reward function for agent } i$$
$$T: S \times A \to PD(S)$$

T is the state transition function, which maps the current state of the environment and the joint action of all agents into the set of probability distributions ($PD$) over the state space S.

A is the joint action space $(A_1 \times \ldots \times A_n)$.

The objective for each agent i is to find a strategy $\partial_i: S \to PD(A_i)$ mapping the joint state S into a probability distribution over the individual action space, which would maximise the reward for that agent over some period. In the stochastic game framework, multiple agents are selecting actions and the next joint state and individual rewards depend on actions of all agents [149]. While assuming that the environment is stationary, Stateless games focus on dealing with multi-agent interactions, Markov game techniques deal with both multi-agent interactions and a dynamic environment. Furthermore, the type of information used by the controller agent for learning also varies across both types of games.

***Independent learners (IL)*** only learn based on their reward observation, on the other hand, ***joint action learners (JAL)*** also apply observations or feedback of actions and possibly rewards of the other agents [131].

### 3.2.2 MARL for deterministic policy control

Reinforcement learning (RL) guarantees optimal policy convergence provided that the agent can sufficiently experiment and the environment in which it is operating is Markovian. RL problems are solved using MDPs without explicitly specifying transition probability, which is needed in value and policy iteration. All reinforcement-learning algorithms share the same goal: to solve sequential decision problems through trial-and-error interactions with the SDN environment [150]. In theory, a reinforcement Learning (RL) agent learns by interacting with its dynamic environment and the goal of the controllers' in RL is to learn a deterministic policy, which maps states to actions. In the ***multi-controller SDN domain***, multi-agent learning is an appropriate solution because of *scalability (complexity)* and inherent *decentralised control*.

Multi-agent reinforcement learning (MARL) is an application of RL techniques to stochastic games. The application of MARL with robustness against adversarial controllers in this thesis will focus on joint policy search, which involves policy control via stochastic optimisation. ***Value function methods*** (unlike policy search methods) need not maintain a representation (explicitly) of policies, instead the aim is to learn a value function that returns the expected cumulative reward for optimal policy from any state. The focus on value functions in relation to RL is to learn value (Q-value) functions though experience. Temporal Difference (TD) method is one of the most common method to learning value functions. ***Policy search methods*** [151] are classified as methods that keep an explicit representation of policies and modify them through a host of search operators. Policy-based methods [152] for multi-agent reinforcement learning searches for policies that optimise an appropriate multi-objective function.

To accomplish our objectives, experiment(s) using an agent-based testbed would be designed to derive data that would be used to evaluate the learning algorithms for controller agents. A set of experiments meant to evaluate the effectiveness of using learning controller agents (on multiple levels) for FEC orchestration would be conducted.

**Deterministic Policy evaluation and validation.** Essentially, this research would design and implement testbeds in some Inter-edge computing and fog environment. Considering this, Cloudsim [33] provides a framework for modelling and simulating cloud computing infrastructure and Services. In later chapter (Chapter 6) of this thesis, description of the testbeds designed based on the CloudSim toolkit would be used to validate the proposed fair resource allocation mechanisms.

## 3.3 Mechanism and controller design

**Multi-agent reinforcement learning (MARL).** A resource allocation game where coalitions or a group of players enforces cooperative behaviour is called cooperative (coalitional) game. An example of a cooperative game is the coordination games, games with multiple pure Nash equilibria in which players choose corresponding (the same) strategies. With cooperative games, software controller agents are willing to form clusters that maximise social welfare, notwithstanding their share of the coalition worth. In such situations, reward distribution becomes trivial, and the main concern is, learning allocation policies that maximise social welfare [129].

The main desirable features of the novel hybrid RAMs [32] as proposed in ***Figure 3-1*** are (1) robust optimisation, (2) an incentive compatible (IC) mechanism that ensures load balancing for performance isolation, group strategy proofness and Pareto optimality or Second-best Pareto optimality (SBPO) in the case of the flat SDN control plane. The proposed fair RAMs consider individual rationality (participation constraint), information requirement, incentive compatibility (IC), and the ability to rank social welfare between the set of possible polices – Vickrey-Clarke-Groove (VCG) mechanisms.

Using reverse games, we apply cybernetic and homeostasis principles of positive and negative (distributed) feedback to develop and analyse hybrid RAMs for *rejuvenation* of the *faulty or "ageing*" distributed SDN control plane. The proposed hybrid scheduling schemes apply coalition game models to implement MARL with robustness against adversarial controllers. The MARL scheme comprises logically centralised DRF policy control mechanism applying physically distributed node offloading strategy feedback to adapt the behaviourally complex (ageing) federated edge and hierarchical fog (dynamic) wireless network.

**Conceptual representation of physically distributed SDN control plane.** An N-person resource allocation (or stochastic) game in coalitional form is defined by the pair $(\mathrm{N}, \mathrm{v})$ where $\mathrm{N} = \{1,2,3, \ldots, \mathrm{n}\}$ is the set of singleton controllers (see Figure3-2) as DRF clusters and v is a real vector (**resource availability**) function, called the characteristic function of the joint offloading games, is defined on the set, $2^N$ of all coalitions or DRF clusters (subsets of N), and satisfying the following:

$(i) \quad v(\emptyset) = 0$

$(ii)(supperadditivity) if\ S\ and\ T\ are\ disjoint\ coalitions\ (S \cap T = \emptyset), then$

$v(S) + v(T) \leq v(S \cup T).$

The definition makes two assumptions: (i) the empty set has a value zero and (ii) the value of disjoint DRF clusters (coalitions) is at least as great when they cooperate as when they do not. The second assumption is not required for some applications of the theory of coalitional games.

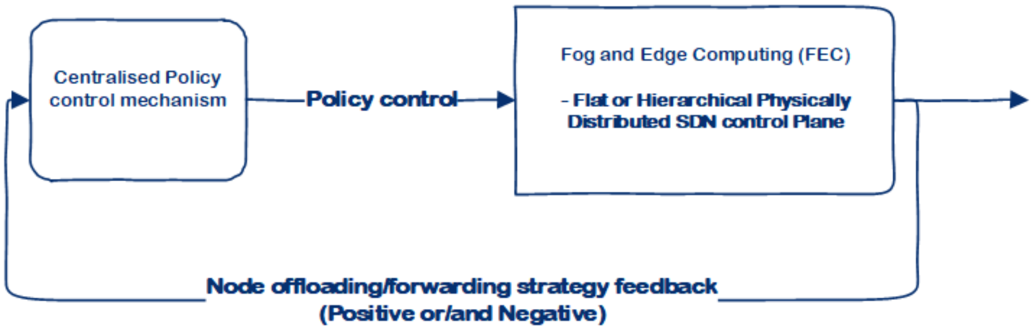



Figure 3-1: Hybrid scheduling schemes (Robust MARL against adversaries) for Joint offloading/forwarding (closed loop control)

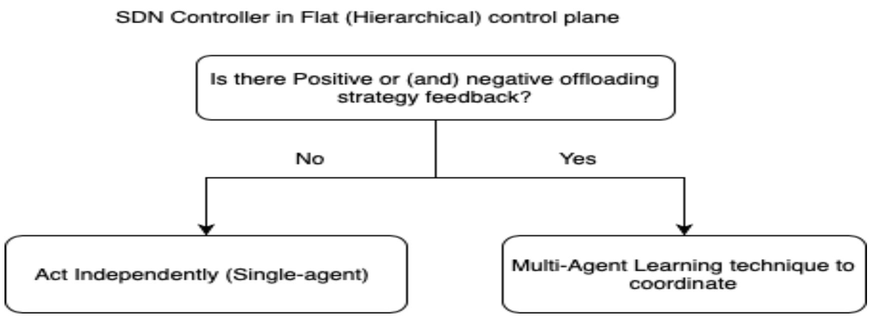



Figure 3-2: Singleton controller design in distributed SDN control plane: IL & JAL [131] for Learning on multiple levels.

### 3.3.1 Study I: Monotonic scheduling scheme

#### A. PFG Model for flat SDN control plane

**The Federated edge and internal dynamics**. The multi-controller platform for edge offloading thrives on the ability to maintain attraction in multiple two-sided markets (***edge-clouds*** or EC providers and offloading users) of the architecture. Social factors and economic encounters between provider (controller) agents may constitute 'network effects' [153, p. 2] or externalities (in value), leading to the notion of 'socio-economic value'. The socio-economic or resource availability value is determined from concurrent ***integrative and distributive***

bargain strategies of data centre (or EC) agents in competitive markets – the incentive compatible mechanism. In federated edge market states, cross-side or indirect network effects due to competitive bargaining, are either positive or negative [153]. As the multi-sided platform for edge offloading grows and with (ECs) applying a production function to provision for bids, many additional task offloading consumers may decide to free ride [154]. Free-rider states occur whereby an increase in non-excludable resources from integrative and distributive EC strategies at the edge is an incentive. This means positive externalities from competitive EC bargaining along with negative can adversely affect SLOs causing platform failure (ageing) due to behavioural complexity in the incentive compatible mechanism – as such, fair allocation policies [155] must consider node offloading strategy feedback to guarantee population (negative) and resource (positive) monotonicity.

Coalition formation is typically modelled in Multi-agent systems using characteristic function games (CFGs), in which the performance of any coalition is not dependent on other coalitions in the system. To model the resilient architecture of flat SDNs, PFG allow for controller-to-controller bargain externalities and these are captured by writing the resource availability value $v$ as a function of DRF cluster and allocation policy (partition) containing this logical cluster as a member. Given a set of ***physically distributed SDN controller nodes*** $A = \{1,2,3, \dots, n\}$, in CFGs any logical DRF cluster $S \subseteq A$ generates a resource value $v(S)$ independent of what other clusters has formed. Applying partition form game (PFG) framework, the effectiveness of one DRF cluster may be affected by the formation of other distinct DRF cluster or coalition [24], [156]. A PFG generates a non-negative integer value $v(S; CS)$ – the resource availability value, where $CS$ is a Joint DRF policy (coalition structure) of A and logical DRF cluster $S \in CS$.

### B. Homeostasis: Distributed node offloading strategy feedback

**Resource Non-monotonicity - Positive Controller-to-controller bargain externalities.** For any mutually disjoint DRF clusters $C, S, T \subseteq A$ in scheduling games for joint offloading and for a partition $\tau$ of $A - S \cup T \cup C$. If the ***resource availability*** value $v(S; \{C \cup T, S\} \cup \tau) >$

$v(S;\{C,S,T\} \cup \tau)$ , then the PFG model for the flat SDN control plane is said to have positive Controller-to-controller bargain externalities.

**Population Non-monotonicity - Negative Controller-to-controller bargain externalities.** Similarly, in games for joint offloading, if resource availability value $v(S;\{C \cup T, S\} \cup \tau) < v(S;\{C,S,T\} \cup \tau)$ a PFG for the flat SDN control plane is said to have negative Controller-to-controller bargain externalities.

To summarise, when a merger between two DRF clusters makes the performance of other clusters better (worse) off, the corresponding scheduling game for joint offloading has positive (negative) externalities [24].

### 3.3.2 Study II: fog link scheduling scheme with bottleneck fairness

#### A. CCF Model for Wireless hierarchical SDN control plane

We apply the constrained coalition formation game [134] model to implement interference-awareness for joint forwarding at the wireless hierarchical SDN control plane as a tuple $\mathcal{G} = \langle A, CS, v \rangle$ where:

- DRF-based software controller agents $A = \{a_1, \dots, a_n\}$ - the set of link nodes for monitoring and feedback using northbound and southbound interfaces.
- $CS \subseteq \Pi(A)$- the set of feasible gateways and low-level controller's placement policies.
- $v$: $(\cup_{C \in CS} \{C\} \rightarrow \mathbb{R})$ is the characteristic (or resource availability) function, which assigns real value(s) to every DRF cluster that appears in some feasible joint gateway and low-level controller's placement policy.

Feasibility of allocations is defined for fog gateway and low-level controller's placement (joint) policy rather than individual DRF clusters. It follows that the constraint on fog gateway and low-level controller's placement (joint) policy implied by $CS$ can be reduced to constraints on individual DRF clusters. A forwarding game $\mathcal{G} = \langle A, CS, v \rangle$ is locally constrained if there exists a set of DRF clusters $C \subseteq 2^A$ such that $CS \subseteq \Pi(\mathrm{A})$ $and$ $CS \subseteq C$ - we refer to the coalitions in $C$ as feasible DRF clusters.

**Representation**. A fog link scheduling is interference-aware (or called valid) if a scheduled transmission on a link flow (channel) X → Y will not result in a collision at either node X or node Y (or any other bottleneck node) due to the simultaneous transmission of other links. Quality of service (QoS) constraints arise from joint computation offloading for IoT in Wireless hierarchical fog networks. For the proposed gateway and low-level controller's optimisation framework, fairness constraints are introduced for **interference-awareness** and mapped using propositional or logic truth whereby Boolean variables of dynamic controller strategies corresponds to link nodes in A. Particularly, the set of Boolean feature constraints $\mathrm{B_A} = \{\beta_i : a_i \in \mathrm{A}\}$. For every router **node $a_i$** there exists a **logical constraint $\beta_i$. Furthermore, let $\Phi$** be the feature selection formula defined over $\mathrm{B_A}$ for gateway and low-level controller's placement (propositional) and built using the logic connectives $(\wedge, \vee, \neg, \vDash)$. $\mathrm{C} \vDash \Phi$ means DRF cluster C satisfies controller feature selection constraint $\Phi$ for joint gateway and low-level controller's placement, if $\Phi$ is satisfied under the ground truth assignment such that $\forall\beta_i$ with $a_i \in \mathrm{C}$ to true and $\forall\beta_i$ with $a_i \notin \mathrm{C}$ to false. In addition, it can be said that a fog gateway and low-level controller's placement (joint) policy or strategy profile CS satisfies $\Phi$ : if $\mathrm{C} \vDash \Phi \; \forall \mathrm{C} \in \mathrm{CS}$.

### B. Homeostasis: Hierarchical (link) node forwarding strategy feedback

**Fog scheduling with bottleneck fairness** - The CCF-based game model can be specified by fairness constraints using pairwise (positive and negative) node forwarding strategy feedback:

- Positive constraints set $\mathcal{P} \subseteq 2^A$ such that a conceptual DRF cluster $C$ satisties a hierarchical interference-aware constraint $P \in \mathcal{P}$ , $\mathcal{P} \subseteq C$;
- Negative constraints set $\mathcal{N} \subseteq 2^A$ such that $C$ satisties a hierarchical constraint $N \in \mathcal{N}.if\;\; N\, is\, not\, a\, subset\, of\; C$;

Note that negative constraints imply respective controller nodes should not work together and as such is handled conjunctively, that is, every cluster must satisfy negative constraints.

Adding DRF cluster size constraint denoted by $S$, a basic forwarding game $\mathcal{G} = \langle A, \mathcal{P}, \mathcal{N}, S, v\rangle$ can be defined as:

Where $A = \{a_1, \dots, a_n\}$ is the DRF agent set, $v: 2^A \to R^N$ is a resource function on $A, \mathcal{P}$ $and$ $\mathcal{N}$ are sets of subsets of $A$ and $S \subseteq \mathbb{N}$. A DRF cluster $C \subseteq A$ is feasible for $\mathcal{G} = \langle A, \mathcal{P}, \mathcal{N}, S, v\rangle$, if:

1. $P \subseteq C$ for some $P \in \mathcal{P}$;
2. $\neg(N \subseteq C) for\ all\ N \in \mathcal{N}$
3. $|C| \in S$.

## 3.4 Design rationale

### A. Centralised controller

**Asset Fairness**. Asset Fairness then tries to equalise the aggregate resource value allocated to each user. Asset Fairness computes for each user i the aggregate share $x_i = \sum_j S_{i,j}$ where $S_{i,j}$ is the share of resource j given to user i. It then applies Max-min across users' aggregate shares, i.e., it repeatedly launches tasks for the user with the minimum aggregate share. While being the simplest scheduling policy mechanism to provides Pareto efficiency, asset fairness violates several important properties: *load balancing for performance Isolation, bottleneck fairness, and resource monotonicity*.

**Dominant Resource Fairness.** For every offloading user, DRF [6] computes the share of each resource allocated to that user in a cluster. The maximum among all shares of a user is called that user (FEC) application dominant share, and the resource corresponding to the dominant share is called the dominant resource – however, different offloading users may have different dominant resources. In certain situations, it may be necessary to allocate more resources to prioritised applications or to users that have contributed more resources to the cluster. This can be achieved via **weighted DRF**, a generalisation of both DRF (**see Appendix A1**) and weighted max-min fairness. In weighted DRF, each user application is associated a weight vector $W_i = < w_{i,1}, \dots, w_{i,m} >$ where $w_{i,j}$ stands for the weight of user applications for resource j. The limitation of DRF is that it provides no mechanism to guarantee resource monotonicity and it is not easily scalable to distributed application architectures.

### B. Decentralised controller (Multi-agent) learning

**Microeconomic theory (Competitive Equilibrium Equal Incomes - CEEI).** GT was first discussed by J. von Neumann and O. Morgenstern in which they developed different methods for finding solutions to two-person zero-sum games. Each player in non-cooperative games takes actions and the result of the game is described by the action taken by each player as well as the value that each player achieves. Wherein cooperative games take into consideration the set of joint actions that any group of players can choose, and reward depends on the group formed and the joint action taken by them and refer to coalitions. While CEEI policy guarantees ***load balancing for performance isolation***, envy-free and Pareto efficiency, it turns out that Nash bargaining solution is not *group strategy (or fault) proof*. Intuitively, this is because CEEI policy assumes a perfectly competitive market that achieves market clearance, i.e., matching of supply and demand and allocation of all the available resources. This policy can lead to giving much higher shares to user applications that use more of a less-contended resource to fully utilise that resource. Thus, a user application can claim more of some underutilised resource to increase overall share of resources to itself.

**Justification of Research methods.** Dominant resource fairness (DRF) guarantees load balancing and strategy proofness however, it is a centralised scheme – not easily ***scalable*** to distributed application environments. Nash Bargaining (or CEEI) policies for the ***distributed problem*** satisfies several fairness characteristics e.g., sharing incentive and Pareto-optimality. However, group strategy proofness, an important property for FEC Data centres is not satisfied which may lead to *scheduler manipulation (or byzantine faults).*

**Software cybernetics (and homeostasis) to design Incentive compatible mechanisms.** In his seminal work in the field, Wiener [157] described cybernetics as the study of control and communication in animals and machines. Cybernetics [157] lays a strong emphasis in the role of *positive and/or negative feedback mechanisms***.** It is transdisciplinary and relevant to mechanical, physical, biological, cognitive, and social systems. Although many aspects of resource allocation games for joint offloading in edge and fog computing have been studied, however, the MARL methodology for multi-controller rejuvenation combines centralised DRF scheduling policy control with positive and negative node offloading strategy feedback related to homeostasis. The application of MARL to provide scalability and robustness in one study distinguishes this research from the other work.

## 3.5 Rejuvenation of decentralised controller placement

We propose to model hybrid monotonic scheduling schemes for priority-based joint offloading as stochastic games with cybernetic and homeostasis principles of feedback (i.e., positive, and negative) to address ***fault-tolerance and scalability*** for the physically distributed *flat and hierarchical* SDN control plane. The novel **self-adaptive** architecture for rejuvenation of decentralised controller deployment adopts MARL implemented as logically centralised scheduling policy control mechanism and applying physically distributed node offloading/forwarding strategy feedback. The closed loop policy control signal from the centralised SDN controller for decentralised controller reconfiguration is highly dependent on homeostasis or node offloading strategy (positive or/and negative) feedback from FEC environment.

### 3.5.1 Study I: Monotonic scheduling policy mechanism

In the first study, we propose to design Integer Partition-Dominant Resource Fairness (IP-DRF) scheduling scheme for priority-based joint offloading to rejuvenate decentralised controller deployment at the flat SDN control plane of the federated edge. The design of the IP-DRF scheme considers: 1) a hybrid scheduling policy mechanism design comprising a logically centralised policy control mechanism with distributed node offloading strategy feedback 2) group strategy-proofness and monotonic scheduling for joint offloading: population and resource.

**Fault tolerant lexicographic multi-objective optimisation**. IP-DRF generalises the Integer partition algorithm [158], [133] by applying lexicographic ordering method to guarantee group strategy proofness and performance isolation via SBPO. In addition, IP-DRF allocation policy generalises DRF [6] and DRFH [85] to distributed cloud data centre environments as well as satisfying resource monotonicity (mitigation of free riding behaviour) by nullifying incentives that increase non-excludable resources thereby avoiding degradation.

### 3.5.2 Study II: Fog bottleneck scheduling policy mechanism

Dfog-DRF is our proposal to provide a decentralised fog controller deployment that generalises weighted DRF [6] or Max-min fair allocation [159], [160] to address bottleneck fairness in distributed fog micro data centres under wireless QoS constraints.

**Fast constraint-based optimisation.** In contrast to previous work [161], [107], [118], [119], we consider proactive rejuvenation of decentralised controller deployment (i.e. both centralised and distributed properties) applying node offloading strategy feedback to optimise cross-layer throughput in fog gateway and low-level controller's placement. However, run-time optimisation plays a central/important role in the optimisation of fog networks. The time available for executing the algorithm is severely restricted and as such, optimisation algorithms need to be fast/quick as well as uphold important fairness attributes [162].

**Throughput-awareness using lexicographic multi-objective optimisation**. To reduce the problem complexity from QoS constraint dimensionality, the proposed game-based methodology applies a divide and conquer method [134] to enable structured (recursive) search through ordered (sorted) DRF cluster lists. Finally, optimisation of fog gateway and low-level controller's placement (joint) policy using lexicographic ordering method is applied to guarantee throughput awareness as well as group strategy proof and Pareto-optimality for the scheduling problem.

## 3.6 Critical appraisal

In joint action learning methodology, the product space of the set of SDN environment states and action sets of the different controller agents influences controller learning. The main drawback of such approach is the large state-action spaces when the number of controllers, SDN environment states, and actions of controllers increase. Furthermore, a shared joint action space approach for multi-controller learning is not always applicable. Particularly, in SDN environments with incomplete information where it may not be possible to fully observe the offloading strategies or actions other controllers take.

A. Large state-action spaces

**Centralised policy control framework - Decrease and conquer approach.** The IP control algorithm proposed in [13], [133] applies an integer partition representation of the search space that groups resource structures into subspaces based on the sizes they contain. Using this representation, it is possible to establish bounds (upper and lower) on the outcome (performance) of the best resource structure in every subspace $P_I{}^A$ of $I \in I_n$ through search. More precisely, let $Max_s$ and $Avg_s$ be the maximum and average outcomes of DRF coalitions of size $s$ ($s \leq n$). For all Integer partitions $I \in I_n$, it is possible to compute the weighted upper bound $UB_I = \sum_{s \in I} Max_s$ and lower bound $LB_I = \sum_{s \in I} Avg_s$ respectively. These bounds are then used to guarantee worst case scenarios on the quality of the best resource allocation structure found so far and to prune unpromising subspaces. For the remaining subspaces, these are explored sequentially until a value is found that is greater than the upper bound of other subspaces; at this point, the algorithm abandons searching such subspaces. The methodology proceeds by performing partial search to establish bound on the optimal resource structure in Inter-edge s by observing the maximum (best) outcome of each resource coalition $C$ in some coalition structure.

**Centralised policy control framework - Divide and conquer approach.** The rejuvenation of decentralised controller deployment for fog gateway and low-level controller's placement makes use of a divide and conquer (D&C) transformation process that helps to avoid the costly possibility of going through every possible DRF cluster formed applying distributed node offloading strategy feedback. This works by splitting the set of feasible clusters into ordered lists. Essentially, the MARL algorithm fills the lists and computes the upper bound of the list from the values of clusters in every list. It builds the feasible fog gateway and low-level controller's placement (joint) policy by adding clusters in sequence. To fasten the exploration for optimal gateway and low-level controller's placement, the methodology applies branch and bound technique to determine the throughput bounds of exploring a set of policies or structures.

B. Incomplete information

A standard classification criterion for multi-agent learning (MAL) is a controller (learning) agent's degree of knowledge of the other controllers in the distributed SDN control plane and their learning processes. The classification results in characterisations of the distributed SDN environment that range from fully unaware to fully aware. Whereas overall stability (learning goal) may be achievable with very little or no awareness of the other controllers' behavior, the behavioural adaptation of the SDN control plane may only be achievable with high or full awareness.

## 3.7 Policy optimisation

***Complexity versus Accuracy.*** In contrast to dynamic programming, the solution quality to the coalition structure generation (CSG) problem improves monotonically with computational time when applying the anytime algorithm [129]. Also, being anytime makes the algorithm strong or robust against failure. The anytime algorithm can further be classified as approximate and exact methods.

### 3.7.1 Approximate method

This class of Algorithms provide "good" solutions to the coalition structure generation problem and have the added advantages of being relatively fast in comparison to other algorithms. This main advantage of this class of algorithms is its ability to scale up with the number of agents and return solutions anytime. Sen and Dutta [163] applied genetic algorithms to the coalition structure generation problem. Their algorithm begins with an initial set of candidate solutions (e.g., a set of coalition structures) and then gradually evolves towards better solutions. The algorithm ***does not provide a guarantee*** on finding optimal solutions nor do they provide worst-case bounds on the quality of the solution they provide. Shehory and Kraus [164] also developed an algorithm that belongs to this class. Their algorithm operates in greedy and a decentralised manner. The search method proposed include adding constraints on the size of feasible coalitions. The solutions provided by Shehory and Kraus [164] algorithm is guaranteed to be within a bound from the optimal but is not guaranteed to find an optimal solution.

### 3.7.2 Exact method

Anytime exact algorithms on the other hand, are guaranteed to find the optimal coalition structure. A generalised approach to the CSG problem is described broadly as a search in an undirected graph (**see Figure3-3**) called the coalition structure generation graph [165]. An alternate representation of the CSG graph is the integer partition (IP) graph [133]; it is based upon the integer partition of a positive integer, which equals the aggregate number of autonomic nodes in the Inter-edge computing environment. This research would focus on developing mechanisms using anytime algorithms/dynamic programming (DP) to solve the coalition structure generation problems for priority-based joint offloading/forwarding at the distributed SDN control plane.

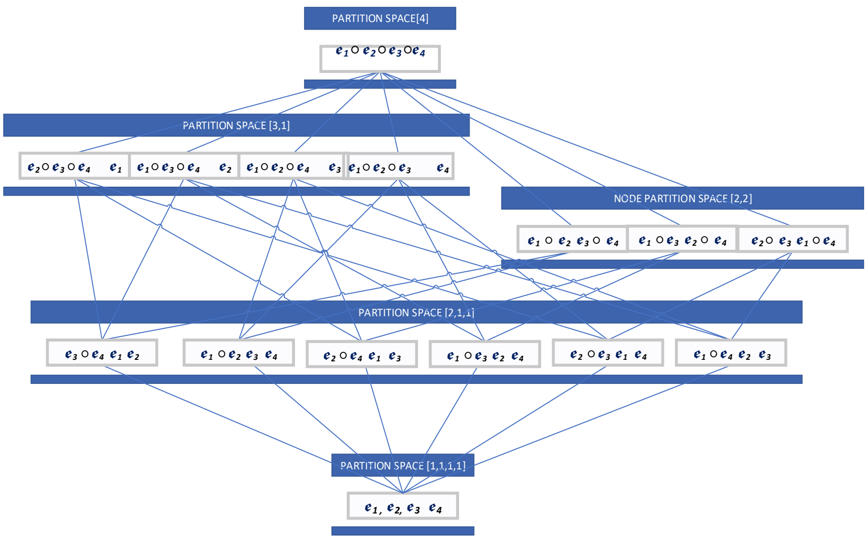



Figure 3-3 CSG graph - DRF cluster mergers & splits (with IP-based subspaces)

## 3.8 Summary

This chapter describes from reverse game theory and software cybernetic perspective, novel hybrid schemes used to solve priority scheduling or coalition structure generation (CSG) problems (i.e., monotonic scheduling and bottleneck fairness) for ***joint offloading/forwarding*** at the federated edge and wireless hierarchical fog networks. The proposed centralised policy mechanisms rely on cybernetic and homeostatic principles of positive or/and negative distributed node offloading strategy feedback to adapt the physically distributed SDN control plane of FEC.

- ✓ First, we discuss alternative mechanisms for priority scheduling of joint offloads – Asset fairness and CEEI.
- ✓ Theoretic principles of games (stochastic) are applied to design direct policy search mechanisms that enable multiple autonomic controllers in cooperation or competition to achieve common goals or overcome conflicting objectives.
- ✓ FEC environment is typically characterised by resource sharing through interworking (cooperation) and competition between multiple controllers (nodes).
- ✓ The exact approach to CSG problems is adopted for control design of priority scheduling and allocation using specialised classes of coalition games namely games in partition form and constrained coalition formation.
- ✓ The proposed fair resource allocation mechanisms focus on joint DRF policy control via application of a MARL methodology based on 'trial and error'.

# Chapter 4 Monotonic scheduling scheme for joint offloading at the federated edge

- This chapter proposes IP-DRF scheme, an incentive compatible mechanism design (self-adaptive) for monotonic and maximum throughput scheduling of multi-resources and multitask IoT applications applying distributed node offloading strategy feedback.
- The flat and physically distributed control plane scheme addresses behavioural complexity of controller or data centre agents in priority-based joint offloading (multitask & multi-resource) via a generalisation of the integer partition (IP) control algorithm to provide SBPO.
- The IP-DRF, a hybrid scheduling scheme for priority-based joint offloading is designed to be robust against “ageing” or ***population and resource non-monotonicity*** due to network externalities. The hybrid scheduling scheme (IP-DRF) is an application of MARL (with adversaries) to ensure demographic parity and satisfy several desirable qualities as compared to asset fair and Nash bargaining allocation policies.

## 4.1 Background

**Fair allocation policies for priority-based joint offloading**. Multi-objective integer programming (centralised) model for asset fair policy is the simplest and ensures Pareto-efficiency [166]; however, the asset fair policy does not guarantee load balancing or sharing incentive and resource monotonicity [6] properties and the model also suffers from dimension complexity. Despite the amount of research on intelligent Inter-edge resource scheduling and allocation, most fair policies primarily focus on strategic games in non-cooperative form, the distributed methodology for the bargaining problem. Competitive equilibrium effort incomes (CEEI) or Nash bargaining allocation policy in the federated edge architecture with market failures is not robust as it violates population and resource monotonicity. For free or competitive markets, it is uncertain whether controller agents applying only marginal integrative value for the offloading users under consideration will move the entire platform closer to the Pareto optimum (PO), unless the optimum conditions are met in the rest of the economic system [23], [167]. In general, the latter will not be the case; Nash bargaining policies [6] violate incentive compatibility or group strategy proofness. In addition, CEEI [6] policy may not necessarily be Pareto-optimal in the case of Inter-edge resource allocation mechanisms seeking to correct for competitive controller-to-controller bargain externalities at the federated edge.

**Decentralised control**. Multitask offloading at the federated edge such that no user is better off if resources are equally partitioned among them is a critical yet unaddressed challenge. The problem implies a reliable and efficient allocation of the federated edge, while maintaining synchronisation between administrative data centre or singleton controller agents. The platform orchestrator for edge offloading can adopt DRF [6], a centralised structure for governance as depicted in Figure3-1. However, the control operations for coordinating the internal control process of individual/administrative ECs for muti-task offloading is distributed. Therefore, weighted max-min fairness or DRF needs to be applicable to distributed EC environments for muti-task offloading at the federated edge. The following properties guide the generalisation of DRF scheduling for Federated edge resources: sharing incentive, group strategy proof, envy-free (embodies the notion of fairness) and Pareto-efficiency as well as population and resource monotonicity.

## 4.2 Reference architecture

The subsection presents the architecture for priority-based joint offloading at the federated edge with two distinct users: administrative data centres or EC s and consumers with distributed multitask applications.

**Interactions at the Federated edge**. For competitive equilibrium, economic encounters (win-win or integrative strategies) between data centre agents can be modelled as a coalition game. At the federated edge, the incentive compatible mechanism is behaviourally complex – parallel win-win and win-lose or distributive bargain strategies to handle IoT task bids for co-offloading. Socio-economic interaction (win-win versus win-lose bargaining) among controller agents for muti-task offloading is modelled as coalition games with externalities.

### 4.2.1 Physically distributed SDN architecture

**Definition 4.1. Multi-sided Platform (and edge offloading nodes).** For the provider side of the two-sided market, datacentre agent set $\mathbb{C}$ represents a set of administrative infrastructure/data centre agents $\mathbb{C} = \{\mathrm{C}_1, \mathrm{C}_2, \ldots, \mathrm{C}_\mathrm{n}\}$. For the offloading consumer side, the set $\mathbb{B} = \{\mathrm{B}_1, \mathrm{B}_2, \ldots, \mathrm{B}_\mathrm{n}\}$ comprises consumer (offloading) agents $\mathrm{B}_\mathrm{i}$, as market coordinator for native IoT task computation offloading.

**Offloading bids for IoT**. Task group list $\Theta_{\mathbb{B}} = \{\theta_{\mathrm{B}_1}, \theta_{\mathrm{B}_2}, \ldots, \theta_{\mathrm{B}_\mathrm{n}}\}$. Each task offloading bid $\theta_j \in \theta_{\mathrm{B}_{1\mathrm{i}}}$ is associated with a task length, number of processing elements (PEs).

Edge resources and resource provisioning. Infrastructure (EC) Host list is defined as $\mathrm{H}_{\mathbb{C}} = \{H_1, H_2, \ldots, H_n\}$. Each Virtual host list corresponds to physical machines $H_i = \{h_{i1}, h_{i2}, \ldots, h_{iz}\}$. Each host (physical) machine $h_{iz} \in H_i$ is associated with compute capacity $cap(h_{iz}) \in \mathbb{R}^+$and a compute speed $Sp(h_{iz}) \in \mathbb{R}^+$. The data centre agent $\mathrm{C}_\mathrm{i}$ uses production (or capacity allocation) function $\mathcal{F}_i(\theta_{\mathrm{B_i}})$ to map consumer bids for joint offloading in administratively controlled physical machine set $H_i$.

**Multi-controller SDN**. The Multi-controller ***(horizontal or flat)*** platform ecosystem, $\mathrm{E} = \{\rho_1, \rho_2, \dots, \rho_n\}$, where each conceptual node $\rho_I$, denotes a parallel market maker and the platform (or broker) node maps the two sides of the federated edge and where $\rho_i: \mathrm{B_i} \twoheadrightarrow \mathrm{C_i}$. Assume each edge broker agent acts as a functional mapping (isolated task workflow) equivalent to DRF scheduling for individual task offloads represented as cloudlet instances [168].

**Definition 4.2. logically distributed DRF clusters** - Using set notation, the set of all possible singleton DRF-based [6] nodes and integrative strategies for main (broker) and secondary controller's placement or DRFH [85] clusters or coalitions for multitask offloading denoted $\hat{\varsigma}$ can be expressed as,

$$\hat{\varsigma} = \{\varsigma \mid \varsigma \subseteq \mathrm{E}, \varsigma \neq \emptyset\} \tag{1}$$

For simplicity's sake, we can rewrite $\hat{\varsigma}$ as $\hat{\varsigma} = \mathrm{P(E)} - \{\emptyset\}$. More specifically,

$$|\hat{\varsigma}| = |\mathrm{P(E)}| - |\{\emptyset\}| = 2^{\mathrm{n}} - 1 \tag{2}$$

Let $\hat{\varsigma}_{\mathrm{r}}$ denote all the possible integrative bargain strategies for broker and secondary controller's placement made up of r nodes. It follows that the number of DRF clusters in $\hat{\varsigma}_{\mathrm{r}}$ can be characterised as the binomial coefficient (or combination) of n and r computed as,

$$|\hat{\varsigma}_{\mathrm{r}}| = \left\langle {\mathrm{n} \atop \mathrm{r}} \right\rangle \tag{3}$$

**Definition 4.3. The joint DRF allocation policy (of East-West interface)**. An exhaustive partition (a strategy vector) of E comprising mutually isolated DRF clusters for muti-task offloading at the federated edge is referred to as Joint DRF policy or more simply, a policy denoted as $\Pi$ and given by,

$$\boldsymbol{\Pi} = \left\{\boldsymbol{\varsigma_1}, \boldsymbol{\varsigma_2}, \dots, \boldsymbol{\varsigma_{|\Pi|}}\right\},\ \mathbf{1} \leq |\boldsymbol{\Pi}| \leq \mathbf{n} \tag{4}$$

In addition to the non-empty subset constraint for constituting DRF clusters, a policy $\Pi$ in the federated edge satisfy the following simple rules:

$\bigcup_{j=1}^{|\Pi|} \varsigma_j = E, \ \forall \varsigma_j \in \Pi$

(ii)$\varsigma_p \cap \varsigma_q = \emptyset, \ p, q \in \{1,2, \dots, |\Pi|\}, \ p \neq q, \ \forall \varsigma_p, \varsigma_q \in \Pi.$

Let $\Pi^E$ denote the set of all Joint DRF policies and the set of policies containing exactly m

Table 4-1: Exponential number of possible DRF clusters and feasible joint DRF policies

| $\boldsymbol{n}$ | 1 | 3 | 5 | 7 | 11 |
|---|---|---|---|---|---|
| $\|\hat{\varsigma}\|$ | 1 | 7 | 31 | 127 | 2047 |
| $\|\boldsymbol{\Pi}^E\|$ | 1 | 5 | 52 | 877 | 678570 |

DRF clusters or coalitions be $\Pi_m^E$. Thus, the number of possible policies is computed as the $n^{th}$bell number thus,

$$|\Pi^E| = \sum_{m=1}^{n} |\Pi_m^E| \tag{5}$$

Where $|\Pi_m^E|$ is evaluated as,

$$|\Pi_m^E| = (1/m!) \sum_{k=0}^{m-1} (-1)^k \left\langle {m \atop k} \right\rangle (m-k)^n \tag{6}$$

Table 4-1 shows for n edge processing nodes exponential growth of the (total) number of possible clusters and feasible Joint DRF policies.

**Definition 4.4. Inter-edge coalition games (ICG)** for co-offloading at the SDN-based federated edge – Stated mathematically, ICG = $(\mathbf{E}, \Gamma)$, where E denotes the controller set in the ICG and $\Gamma$ is a scalarised or weighted function, called the (characteristic) QoS value of the game, and defined on any feasible Joint DRF policy $\Pi$.

**Integrative EC bargaining.** Let $\hat{\varsigma}_{2\leq r\leq n}$ represent all possible Inter-edge coalitions implemented as DRF or DRFH [85] clusters for joint multitask offloading. It can be deduced that$|\hat{\varsigma}_{2\leq r\leq n}| = \sum_{r=2}^{n}|\hat{\varsigma}_r|$, and since$|\hat{\varsigma}| = \sum_{r=1}^{n}|\hat{\varsigma}_r|$, this implies using eq. (2) and (3) that,$|\hat{\varsigma}_{2\leq r\leq n}| = |\hat{\varsigma}| - |\hat{\varsigma}_1| = |\hat{\varsigma}| - \langle{n \atop 1}\rangle = |\hat{\varsigma}| - n$ therefore,

$$|\hat{\varsigma}_{2\leq r\leq n}| = 2^n - n - 1 \tag{7}$$

**Distributive EC bargaining**. In ICGs (win-win or integrative strategies), any DRF cluster $\varsigma \subseteq E$ generates a socio-economic value $\gamma(\varsigma)$ independent of any zero-sum strategies played. However, in a more realistic settings, the competitive marketplace, concurrent Integrative and distributive (non-zero-sum and zero-sum) strategies for allocation of federated edge resources implies (zero-sum) bargain value externalities; a divergence between private and social welfare in joint multitask offloading leads to platform failure. Zero-sum bargain value externalities [24], [169] from cross-side effects of integrative strategies at the federated edge are captured by writing the resource availability or socio-economic value $\gamma(\varsigma;\Pi)$, where $\Pi$ is the policy or strategy profile from E and DRF cluster $\varsigma \in \Pi$.

### 4.2.2 Fault-prone SDN control plane

Integrative bargain strategies for broker and secondary controller's placement can cause cross-side effects or externalities that lead to inefficient allocation for the offloading user application due to parallel distributive EC bargain strategy. Dynamics in the incentive compatible mechanism is monitored from two perspectives: (1) Population monotonicity – from EC (Infrastructure) providers to offloading consumers and (2) Resource monotonicity – from offloading user to EC providers. We analyse complex fairness characteristics for multitask offloading that arise from negative and positive same-side network effects [153], [170] as depicted in **Figure 4-1.**

**Ageing of distributed SDN architecture** – multitask offloading, EC data centres and multiple resource types. Consider a multi-sided platform for edge offloading described as an ICG comprising four (n = 4) data centre agents for $C_1$, $C_2$, and $C_3$, $C_4$ and four offloading consumer agents: $B_1$, $B_2$, $B_3$ and $B_4$. Each platform agent is matchmaker and monitors offloading consumers request for virtual resources in data centre hosts ($i \leq 4$), platform monitoring agent $\rho_i$. Let consumer bids for IoT offloading at the federated edge be of at least two cloudlets [171] or container-based) requirement classes: video stream processing applications $\Theta_{\mathcal{B}}^{CX}$ (bandwidth-intensive) and interactive offloads like health monitoring or intelligent transportation $\Theta_{\iota}^{CY}$. Multiple resources (containers/VMs) are critical to satisfy QoS objectives of low response times: Interactive for example, health monitoring or ITS applications $\Theta_{\iota}^{CY}$ require medium containers that require, for example, six vCPUs, 2GB of memory, 2,000 Mbps bandwidth and 60 GB of storage. bandwidth-intensive offloading application (video processing) $\Theta_{\mathcal{B}}^{CX}$ that require large containers comprising of eight vCPUs, 4 GB of memory, 8,000 Mbps bandwidth and 100 GB of storage.

Administrative data centre (EC) agents $C_1$, $C_2$, $C_3$ have limited host infrastructure capacity (in MIPs/bandwidth) for medium VMs and large VMs and data centre agents $C_4$ only provisions

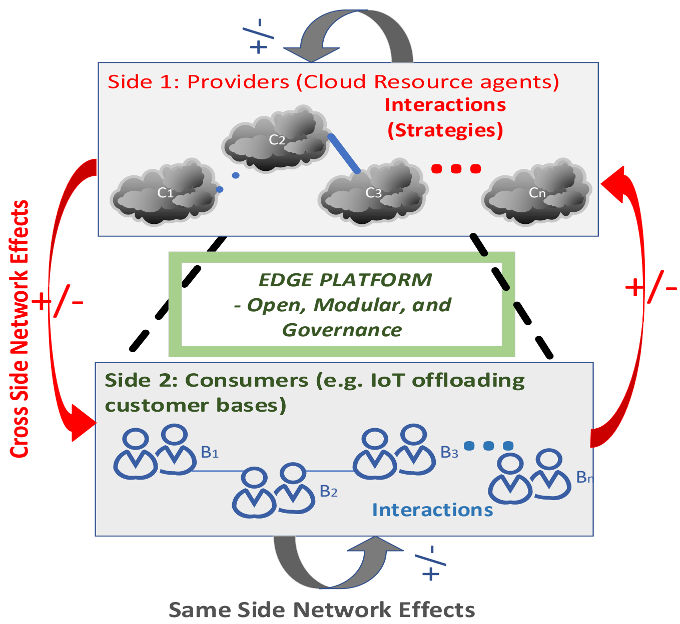



Figure 4-1: Network effects in multi-sided platform for edge offloading: Two-sided markets

large VMs for video stream processing and no medium VMs. In addition, assume the marketplace is a time-shared resource provisioning scheme.

**Incentive compatibility issue I**: Negative Controller-to-controller bargain externalities in the transaction platform – Governing the congestion problem in EC negotiation for multitask offloading at the SDN-based federated edge architecture as shown in Figure4-2.

**Example 4.1**. Population non-monotonicity from concurrent short and long-term Data centre (DC) host failures. Assume at some instance $t$ in platform evolution, data centre agents $C_2$ implements reliability via distributive bargain strategy for excess resource capacity from competitors due to temporary (short-term) physical host outages for bids $\Theta_{\iota}^{CY}$ of consumer agent $B_2$. Whereas data centre agents $C_1$ has 'just enough' compute capacity $cap(h_{1z})$ , data centre agents $C_4$ is largely underutilised for large containers or VMs considering consumer bids at $B_1$ and $B_4$ respectively. Fewer bids are submitted at offloading consumer agent $B_3$ and data centre agents $C_3$ can play host to container-based (or VM) migration workload based on zero-sum bargain strategies to address short-term under-provisioning mistakes. Concurrently, resource outages due to disasters [71], e.g. floods can cause available physical host machines $H_1^{t-1} = \bigcup_K h_{1,K}$ to diminish to $H_1^t = \bigcup_J h_{1,J}$ where $\|J\| \ll \|K\|$ and result in a major decline in its production function $\mathcal{F}_1^t(\Theta_B^{CX}, \Theta_{\iota}^{CY})$ and subsequently excessive demand in large VMs for video surveillance applications at offloading consumer agent $B_1$.

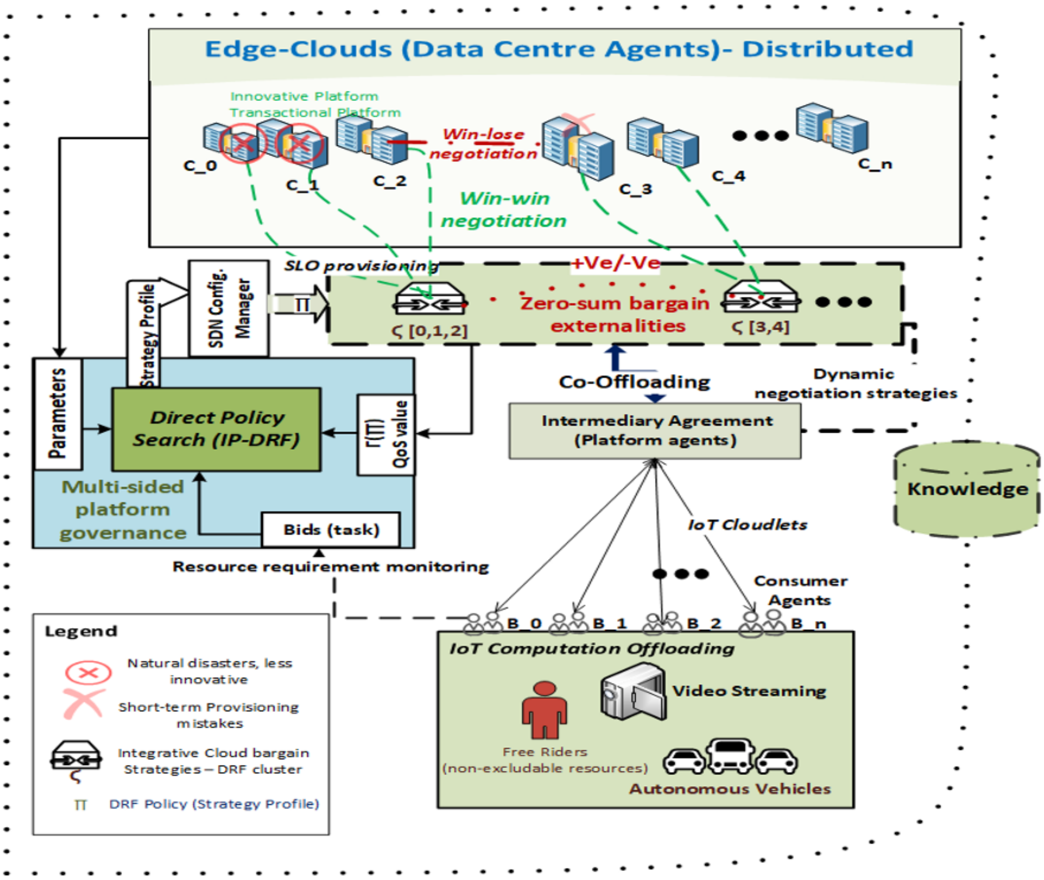



Figure 4-2: IP-DRF (monotonic) scheme for rejuvenation of flat SDN control plane

**Negative distributive bargaining of data centre (EC) agents**. Geographic proximity allows the presence of distributed bargain (for VM migration) strategies between data centre agents $C_3$ and $C_2$ to correct short term infrastructure unavailability. however, in the non-zero-sum bargaining game (i.e., ICG), $C_3$ can adopt integrative strategy or main (broker) and secondary controller's placement and merge in a win-win agreement with resource-deficient provider agent $C_1$. Inter-edge (or DRFH cluster) cluster $\varsigma_{13}$ denoted by $(\rho_1 \circ \rho_3)$ satisfying SLOs for task offloading of bandwidth-intensive application $\Theta_B^{CX}$ constitutes a subgame Nash equilibrium. However, the subgame perfect NE policy may not be Pareto-efficient, this is because integrative EC strategies to form DRF clusters portends cross-side controller-to-controller bargain externalities (or socio-economic $\gamma$ value loss) in DRF cluster $\varsigma_2$. Essentially, some offloading consumers in logical DRF cluster $\varsigma_2$ can suffer increased SLO violation for interactive applications $\Theta_{\iota}^{CY}$. For the priority scheduling of multitask offloading, the proposed allocation scheme considers negative distributive EC bargain externalities to mitigate platform failure.

**Incentive compatibility issue II:** Mitigating the free riding problem from distributive EC bargaining in multitask offloading at the federated platform (innovative) edge.

**Example 4.2**. Resource non-monotonicity from non-excludable resources increase the free riding incentive. A process for the provisioning of edge services proceeds as follows: first, each offloading user application submits a bid $\varphi$ for resources. Then, the edge infrastructure provider maps the vector of bids $\varphi$ into provisioned resources $\mathcal{F}(\varphi)$ and a payment the $p_i$for each multitask offloading user (application) i, we say $\mathcal{F}$ the production function and the $p_i(\varphi)$ the payment function. Assume, the production and payment functions are known in advance. Specifying the space of application bids, the production and payment specify the negotiation process. Free riders in the (distributed) singleton DRF clusters of the platform create problems for allocation of resources particularly when their actions fail to communicate signals to the allocator of resources (i.e., data centre agents). In platforms that support innovative service marketplaces for cross-platform capabilities e.g., multitask offloading in video stream processing, long-term integrative strategies can increase non-excludable resources (i.e., containers or VMs) e.g., health or ITS applications – hence, promoting the free riding incentive.

**Positive distributive EC bargaining**. Assume the incentive for free riding (see **Figure 4-2**) at consumer agent $B_4$ is non-excludable resources that exist because of known positive distributive EC bargain externalities of $C_4$. Data centre agents $C_3$ has enough capacity for large containers or VMs handling bandwidth-intensive applications as well as play host to container (or VM) migration of non-excludable resources (medium VMs or containers) from positive zero-sum bargaining of data centre agents $C_4$. Moreover, in environments whereby integrative or win-win strategies of data centre agent $C_3$ in the ICG can only lead to socio-economic value $\gamma$ gain (an increase in non-excludable resources) for third-party offloads at consumer agent $B_4$ – this only increases the incentive to free ride. Uncontrolled integrative EC strategies in the innovative platform leads to free riding or cross-side negative network effects that result in market failure (underproduction) or increased SLO failures. The CEEI policy may not satisfy QoS unless the cross-side network effect or incentive leading to free riding in the resource non-monotonic environment from positive distributive bargaining of $C_4$ is internalised or nullified. One way to internalise the incentive for free riding in the federated edge is to implement integrative strategies or DRF clusters.

## 4.3 System model

### 4.3.1 Monotonic scheduling games for Joint offloading

#### A. Markovian model.

Extending the framework from chapter 3, the multi-objective optimisation framework to analyse priority scheduling for joint offloads in the intelligent multi-sided platform or flat architecture is defined as a 5-tuple Multi-agent markov decision problem MDP $(\boldsymbol{G}(\Pi^{\mathrm{E}}, \boldsymbol{\xi}), \mathbb{T}, \boldsymbol{Q}^{\pm}, \boldsymbol{\eta}, \boldsymbol{\Gamma})$.

Briefly, $\boldsymbol{G}(\Pi^{\mathrm{E}}, \boldsymbol{\xi})$ represents the space of Joint DRF policies in the federated edge environment, $\mathbb{T}$ is discrete time; $\boldsymbol{Q}^{\pm}$ is a set of federated edge with failure states, that is, ICGs with either strictly positive or negative controller-to-controller distributive bargain externalities. $q_0$ is the initial state; and $\eta$ is a transition function mapping to the next federated edge environment state of resource functions.

$\boldsymbol{G}(\Pi^{\mathrm{E}}, \boldsymbol{\xi})$, the bargaining space for multitask offloading is represented as a graph. Particularly, we say, $G$ is the scheduling policy graph, and $\Pi^{\mathrm{E}}$, is the set of vertices or Joint DRF policy subspaces of $G$, each comprising at least one policy. The set $\xi$ of edges in $G$ represent mergers and splits of DRF clusters - the existence of Integrative and possible distributive EC bargain strategy. The search cost metric that simply computes the cost of searching a logically distributed policy subspace $\Pi^{\mathrm{E}}$ in $G$, equal to the aggregate number of policies in the policy subspace $\Pi^{\mathrm{E}}$.

The discrete time $\mathbb{T} = (\boldsymbol{T}, \boldsymbol{\Sigma}, \boldsymbol{\Upsilon}, <)$ is a dimensionless measured space with strict total ordering. $T$ is a set of time periods with at least one element $\kappa_0$, that is, $\kappa_0 \in T$. $T$ can be viewed as a set of abstract time steps $T = \{\kappa_0, \kappa_1, \kappa_2, \dots\}$. $\Sigma$ is defined as $\sigma$-Algebra over time-period $T$ or more simply as a non-empty set of subsets of $T$ closed under union and complement with respect to $T$. $\Upsilon$ is the measure of peak offload bids for the real-time IoT application in $T$. For example, let $t_1 \in \Sigma$ then $\Upsilon(t_1) = \partial$. We define $\partial$ as the peak number of application instance bids. The 'less than' $(<)$ operator represents a strict total order on $T$.

### B. Homeostatic regulation for monotonicity

The set of federated edge or ICG environment (feedback) states with resource and population non-monotonicity $\boldsymbol{Q}^{\pm} = \{\boldsymbol{q_0}, \boldsymbol{q_1}, \boldsymbol{q_2}, \dots\}$ where $q_0$ is the initial state and *the max-min fairness problem* for each feasible state $q_j$ in $\boldsymbol{Q}^{\pm}$ corresponds to finding the best response Joint DRF policy in the set ${\Pi_{qj}}^{E} \equiv \boldsymbol{G}$. Let $\Pi_{qj}$ represent the allocation policy such that $\Pi_{qj} \in {\Pi_{qj}}^{E}$ and $\boldsymbol{\Gamma}(\Pi_{qj})$ is the weighted sum (QoS value) of the Joint DRF policy or strategy profile $\Pi_{qj}$.

The State transition function $\boldsymbol{\eta}: \boldsymbol{T} \rightarrow \boldsymbol{Q}^{\pm}$ maps a specific time to specific federated edge environment states (with market failure) –population or resource non-monotonicity.

**Definition 4.5: Inter-edge coalition games (ICGs) with positive and negative controller-to-controller bargain externalities – DRF agents** undergo monitoring of edge-data centre agents simultaneously adopting both Integrative and distributive negotiation in Inter-edge resource allocation for reliable co-offloading. The states $\boldsymbol{Q}^{+}$ is modelled as ICGs having strictly positive (possibly weak) bargain externalities if for any mutually disjoint integrative coalitions $\varsigma_1, \varsigma_2, \varsigma_3 \subseteq \mathrm{E}$ and for a partition $\Xi$ of $\mathbf{E} - \{\varsigma_2 \cup \varsigma_3 \cup \varsigma_1\}$, we have, $\gamma(\varsigma_2; \{\varsigma_1 \cup \varsigma_3, \varsigma_2\} \cup \Xi) \geq \gamma(\varsigma_2; \{\varsigma_1, \varsigma_2, \varsigma_3\} \cup \Xi)$. Similarly, states $\boldsymbol{Q}^{-}$ is modelled as an ICG having strictly negative (maybe weak) EC bargain externalities if the resource availability or socio-economic value $\gamma(\varsigma_2; \{\varsigma_1 \cup \varsigma_3, \varsigma_2\} \cup \Xi) \leq \gamma(\varsigma_2; \{\varsigma_1, \varsigma_2, \varsigma_3\} \cup \Xi)$. In simple terms, an ICG has positive (negative) zero-sum bargain externalities if a merger (integrative EC strategies) between two DRF clusters makes the resource availability for some third-party consumer tasks in other logical cluster better (worse) off. Socio-economic (or resource availability) value $\gamma$ to determine the QoS value $\Gamma$ has two dimensions: a non-zero-sum bargain value and zero-sum bargain value.

**Socio-economic value** $\gamma$. The QoS **value** is evaluated as the rate of success (in resource allocation) satisfying strict constraints (latency) for consumer offloading - providing performance isolation for IoT application instances. Performance related to QoS in SLO contracts for IoT applications is considered the most critical objective in offloading at the edge.

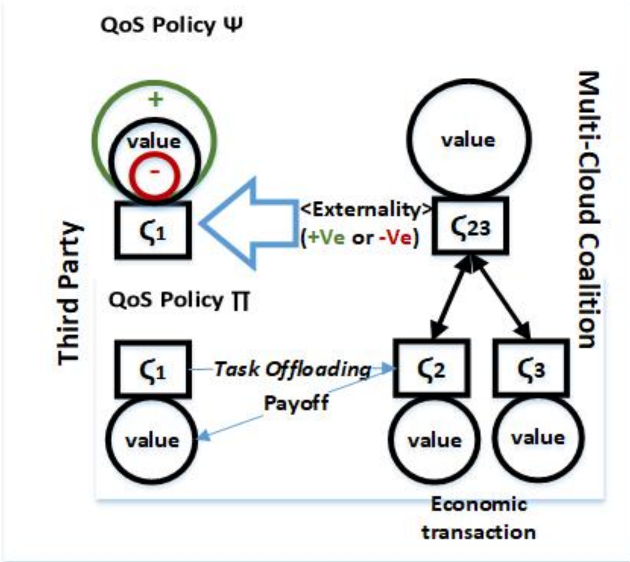



Figure 4-3: Controller-to-controller distributive bargain externality (+Ve/-Ve or 0) for third-party offloads.

**Non-Zero-Sum bargain value**. For any distributed DRF coalition $\varsigma$ in policy $\Pi$ ($\varsigma \in \Pi$) at discrete time $t$, this non-zero-sum bargain (integrative) value is denoted $\gamma_{t,\Pi}(\varsigma)$.

**Evaluating Controller-to-controller distributive bargain externalities (Negative & Positive).** Because a DRF cluster $\varsigma$ can co-exist (in parallel) with other clusters in a Joint DRF policy, suppose $\bar{\varsigma}$ denotes the set of other DRF clusters, e.g., $\bar{\varsigma} = \Pi - \varsigma$. We denote by $e_{t,\bar{\varsigma}}$ as the zero-sum bargain externalities acting upon the QoS value of $\varsigma$ by $\bar{\varsigma}$ in time $t$ (see Figure 4-3). Let $\gamma_{t,\Psi}(\varsigma)$ be the non-zero-sum bargain value for embedded DRF-based coalition $\varsigma$ in some other policy $\Psi$, and $\Pi \neq \Psi$. Now, if for any two DRF coalitions $x, y \in \Pi$, let $x \cup y \in \Psi$, where $\{x, y\} \cap \varsigma = \emptyset$ and $\{x, y\} \cap \bar{\varsigma} \neq \emptyset$. Controller-to-controller bargain externalities if any can now be determined by observing the divergence in 'bargain value' or resource availability for DRF-based coalition $\varsigma$ in policies $\Pi$ and $\Psi$. Externalities acting upon the resource availability (or QoS) value of embedded DRF cluster $\varsigma$ is computed as,

$$e_{t,\bar{\varsigma}} = \gamma_{t,\Psi}(\varsigma) - \gamma_{t,\Pi}(\varsigma) \quad (8)$$

Socio-economic (QoS) loss (likewise socio-economic gain) are said to occur when $\|e_{t,\bar{\varsigma}}\| <$ $(>)0$ – the resource availability for applications that is lost (or gained). The sum of *non-zero-sum* bargain value and (external) *zero-sum* bargain externalities yield the socio-economic interaction value for any logical DRF cluster ς.

$$\gamma_t(\varsigma, \Pi) = \gamma_{t,\Pi}(\varsigma) \pm e_{t,\bar{\varsigma}} \tag{9}$$

**Fault-tolerant control using lexicographic multi-objective optimisation**. A primary characteristic of pareto optimisation for multi-criteria optimisation (MCO) problems is the need for a decision maker's intervention in the optimisation process**.** The resource allocation problem for co-offloading implies maximising resource availability ($f_1$) to guarantee latency requirements in SLOs as well as reducing cost of processing tasks ($f_2$). Therefore, we propose the a priori Multi-Criteria Optimisation (MCO) method called lexicographic method [172], [43] which assumes that objectives can be arranged in a hierarchy of preference or importance. The initial step in this method is to categorise the objective functions into different levels based on their importance. The highest level is the most important whilst the lowest level is the least important. Due to its simplicity and effectiveness, the conventional approach for dealing with the MCO aspect of the problem is to use the scalarisation approach, in which multiple objective functions are combined to a single objective function. Thus, we apply a common scalarisation approach known as the weighted sum method. Accordingly, we describe the bi-objective maximisation problem for QoS control such that the latency $f_1$ is the highest priority or objective whereas the cost $f_2$ is the least priority. Precisely, let $f_1$ and $f_2$ denote the objectives with the highest and least importance respectively.

**Joint DRF policy (or Strategy profile) payoff**. The multi-objective (scalarised) value or QoS value for any policy Π denoted $\Gamma_t(\Pi)$, is computed as the weighted sum of QoS metrics for offloading policy configuration $\mathfrak{E}^{\Pi}$ i.e., for all embedded DRF clusters ς in Π,

$$\Gamma_t(\Pi) = \sum_{i=1}^{m} \alpha_i \cdot f_i(\mathfrak{E}^{\Pi}) \tag{10}$$

$\sum_{i=1}^{m} \alpha_i = 1$ and $f = (f_1, f_2, \dots, f_m)$, where $f_i$ is the numerical approximation of the ith objective from $m$. The joint DRF payoff is defined in order of priority by: (1) the latency of

IoT offloading consumers and (2) the cost of processing in EC providers. Socio-economic interactions $\mathfrak{E}^{\Pi} = \{\mathfrak{E}_{PO}{}^{\Pi}, \mathfrak{E}_{SBPO}{}^{\Pi}\}$ comprises: (1) only win-win or Main (broker) and secondary controller's placement value in joint policy configuration $\mathfrak{E}_{PO}{}^{\Pi} = \sum_{\varsigma \in \Pi} \gamma_t\,(\varsigma)$ as it is assumed $\left\|e_{t,\bar{\varsigma}}\right\| \rightarrow 0$ (2) Main and secondary controller's value and win-lose or distributive bargain externalities in joint policy configuration setting $\mathfrak{E}_{SBPO}{}^{\Pi} = \sum_{\varsigma \in \Pi} \gamma_t\,(\varsigma, \Pi)$ from **definitions 4.4 & 4.5** to model the two different classes of ICGs at the edge. The distributed Joint DRF policy (or strategy profile) payoff $\mathbf{\Pi}^{*}$ is computed as $\mathbf{\Pi}^{*} \leftarrow \boldsymbol{max}_{\boldsymbol{\Pi \in G^n}}(\Gamma_t(\Pi))$.

**Distributed calculation of Socio-economic or resource availability value(s).** The core idea of the distributed coalition value calculation (DCVC) [173] is to represent the space of all feasible DRF clusters for co-offloading in the form of structured lists to support the efficient distribution of the utility calculation among controller agents and virtually no redundant calculations are performed. Given these core definitions/descriptions of the co-optimisation model, we show, in **section 4.4**, how this background is applied to develop the fast Second-best Pareto-optimal (IP-DRF) policies.

### 4.3.2 The joint DRF policy space

The proposed joint policy [149] control approach to govern node strategies adopts a gradient-based method for representation of the policy space $\Pi^{E}$called the Integer partition (IP) [13], [45], [133], [158], [174] graph.

**Joint policy search spaces**. The set of all possible Joint DRF policies, $\Pi^{E}$ in the federated market is grouped into subspaces, each of which is represented by an **integer partition** of $n$, the number of edge nodes in $E$. To describe this further, an integer partition of $n$ consists of integer parts, the sum of which equals $n$.

**Joint policy search based on graph representation**. In the IP graph representation for joint policy space, every vertex in the graph represents a policy subspace that comprises joint DRF policies (also called policies), nodes are categorised into n levels (partition spaces) denoted $I_1, I_2, \ldots I_n$ Partition space $I_k (k \leq n)$ contains the **logical policies** comprising k **DRF clusters** [43] [174] [45]**.** Specifically, if $G^n$ denotes the IP graph of bargain strategies comprising n FEC provider nodes then the vertex or policy subspace $\Pi_{I=[|\varsigma_1|,|\varsigma_2|\ldots,|\varsigma_q|]} \in I_q$, the partition subspace that consists of policies with q distinct or **non-overlapping DRF-based coalitions** such that, $|\varsigma_1| + |\varsigma_2| \ldots + |\varsigma_q| = |E| = n$, and $q \leq n$. For example, **Figure 4-4** shows the IP graph $G^4$ consisting of vertices and the Joint DRF policy subspaces for 4 edge nodes enumerated as,

$$G^4 = \{\Pi_{[4]}, \Pi_{[2,2]}, \Pi_{[3,1]}, \Pi_{[2,1,1]}, \Pi_{[1,1,1,1]}\} \tag{11}$$

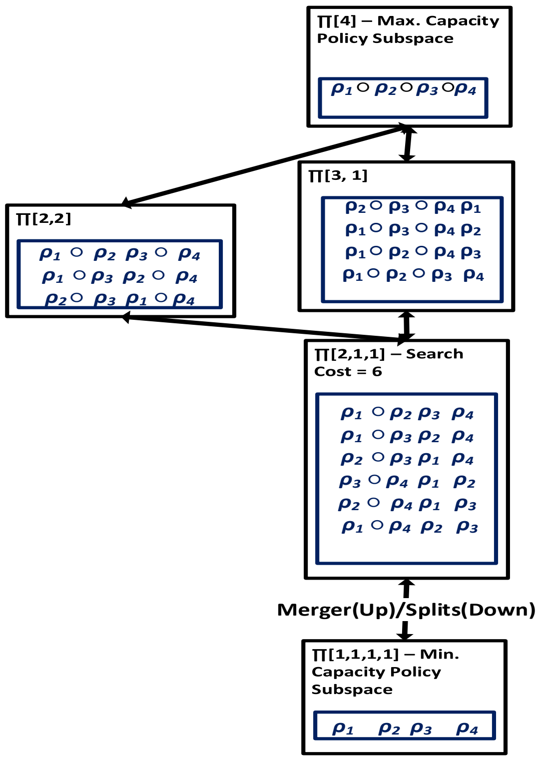



Figure 4-4: Joint DRF policy graph for n= 4 nodes containing lists $L_s$ of all possible Integrative EC strategies of size $s \leq n$.

## 4.4 Robust MARL against adversaries (IP-DRF scheme)

**Multi-objective (and Anytime) optimisation**. For anytime optimisation to effectively govern against the Controller-to-controller bargain externalities at the flat SDN plane of the federated edge, the core approach is to generate an initial joint policy (strategy profile) for muti-task joint offloading that is guaranteed to be within a ratio bound $\beta$ from the optimal joint policy. To implement Pareto improvements in Joint DRF policy and bound $\beta$, more policies are searched/explored.

### 4.4.1 Overview of centralised scheduling policy control

**Lexicographic multi-objective optimisation**. The IP control algorithm is based on an integer partition representation [133], [175] of the search (bargain) space that groups Joint DRF policies into subspaces based on cluster sizes $s \leq n$. Using this graph representation for multi-objective optimisation, our solution follows a branch and bound algorithm design paradigm to establish bounds (upper and lower) on the scalarised value of the best Joint DRF policy in every subspace $\Pi_I, I \in G^n$ through partial search. More precisely, let $Max_s$ and $Avg_s$ be the maximum and average weighted-sum (QoS) payoff of policies of cluster sizes $s$ ($s \leq n$). For all policy subspaces $I \in G^n$, it is possible to compute the weighted-sum upper bound $UB_I^w = \sum_{s \in I} max_s$ and lower bound $LB_I^w = \sum_{s \in I} Avg_s$ respectively. These ***weighted-sum*** bounds are then used to guarantee worst case scenarios on the payoff of the most optimal joint DRF policy (strategy profile) and also to prune redundant joint DRF policy subspaces. As for the remaining logical joint policy subspaces, they are searched one after another unless a weighted-sum value is found that is higher than the upper bound of other subspaces, in which case we abandon searching that joint policy subspace.

### 4.4.2 MARL policy for Monotonic scheduling

**Background**. To implement fault-tolerance and scalability for IP-DRF scheme in the flat SDN control plane based on the weighted-sum upper $UB_I^w$ and lower $LB_{I'}^w$ bounds, several definitions are generalised using theorems provided by Rahwan et al [129], [158], [133].

From Eq. (10), we established that bargain (QoS) payoff $\Gamma$ of a joint policy $\Pi \in \Pi^E$ is the weighted-sum of peformance objectives for distributed DRF clusters $\varsigma$ in $\Pi$. Now, let $\bar{\varsigma}$ refer to the logical DRF clusters in E that do not belong to $\varsigma$ ($\bar{\varsigma} = E - \{\varsigma\}$). More so, we generalise the definition of a joint DRF policy to include sub-policies of $\varsigma$ as a set comprising disjoint main and secondary controller's placement (as DRF coalitions) of which the union equals DRF cluster $\varsigma$, and denote the set of all sub-policies as $\Pi^{\varsigma}$. Every element of a sub-policy $X \in \Pi^{\varsigma}$ is a logical DRF cluster. However, for the sake of clarity, we use small letter chi $\chi$ to denote such logical DRF cluster. For any Joint DRF policy $\Pi$, the payoff of a sub-policy $X \subseteq \Pi$ denoted $\Gamma(X, \Pi)$, is the weighted sum of objectives for all logical DRF clusters in that sub-policy, i.e. $\Gamma(X, \Pi) = \sum_{i=1}^{m} \alpha_i . f_i(\sum_{\chi \in X} \gamma(\chi, \Pi))$.

#### 4.4.2.1 Computing Fault-tolerance bounds (Upper and Lower)

***Definition 4.4.1.*** *The Upper and lower weighted-sum payoff (QoS) bounds of Joint DRF policy subspaces can be computed.*

*Consider an ICG with congestion enviroment states $Q^-$ (likewise free-rider problem $Q^+$) in muti-task offloading at the federated edge, then given any isolated (Integrative) DRF coalition $\varsigma \in \hat{\varsigma}$, a DRF sub-policy $X \in \Pi^{\varsigma}$, and a policy $\Pi \supseteq X$, the following hold where the singleton EC (controller) agents in $\bar{\varsigma}$ are denoted as $\bar{e}_1, \bar{e}_2, \dots, \bar{e}_{|\bar{\varsigma}|}$:*

For population non-monotonicity states ($Q^-$), negative controller-to-controller distributive bargain externalities in muti-task offloading,

$$\Gamma(X, \{\bar{\varsigma}\} \cup X) \leq \Gamma(X, \Pi) \leq \Gamma\big(X, \{\{\bar{e}_1\}, \{\bar{e}_2\}, \dots, \{\bar{e}_{|\bar{\varsigma}|}\}\} \cup X\big)$$

For resource non-monotonicity states ($Q^{+}$), positive controller-to-controller distributive bargain externalities in co-offloading,

$$\Gamma(\mathrm{X}, \{\bar{\varsigma}\} \cup \mathrm{X}) \geq \Gamma(\mathrm{X}, \Pi) \geq \Gamma\big(\mathrm{X}, \{\{\bar{e}_1\}, \{\bar{e}_2\}, \ldots, \{\bar{e}_{|\bar{\varsigma}|}\}\} \cup \mathrm{X}\big)$$

**Explanation.** Let policies $\Pi' = \{\bar{\varsigma}\} \cup \mathrm{X}$ and $\Pi'' = \{\{\bar{e}_1\}, \{\bar{e}_2\}, \ldots, \{\bar{e}_{|\bar{\varsigma}|}\}\} \cup \mathrm{X}$. Also, assume $\Pi \neq \Pi'$ and $\Pi' \neq \Pi''$. Then, given an ICG with strictly negative and positive controller-to-controller bargain externalities, it can be proven that QoS value $\Gamma(\mathrm{X}, \Pi') \leq (\geq) \Gamma(\mathrm{X}, \Pi)$ and $\Gamma(\mathrm{X}, \Pi) \leq (\geq) \Gamma(\mathrm{X}, \Pi'')$.

Essentially, definition 4.1 bounds the payoff (i.e., weighted sum reward) of any given sub-policy of DRF coalition $\varsigma$. For every sub-policy $\mathrm{X} \in \Pi^{\varsigma}$, the (weighted sum) upper payoff bound ${UB_w}^{\mathrm{X}}$ and lower payoff bound ${LB_W}^{\mathrm{X}}$ can be computed in ICGs with congestion (or free rider) states as,

$${LB_w}^{\mathrm{X}}\left({UB_w}^{\mathrm{X}}\right) = \sum_{i=1}^{m} \alpha_i . f_i \left(\sum_{\chi \in \mathrm{X}} \gamma(\chi, \mathrm{X} \cup \{\bar{\varsigma}\})\right) \quad (12)$$

$${UB_w}^{\mathrm{X}}\left({LB_w}^{\mathrm{X}}\right) = \sum_{i=1}^{m} \alpha_i . f_i \left(\sum_{\chi \in \mathrm{X}} \gamma\left(\chi, \mathrm{X} \cup \left\{\{\bar{e}_1\}, \{\bar{e}_2\}, \ldots, \{\bar{e}_{|\bar{\varsigma}|}\}\right\}\right)\right) \quad (13)$$

Assuming $\mathrm{X} = \{\varsigma\}$, then we can compute the weighted-sum upper payoff bound ${UB_w}^{\varsigma}$ and lower bound ${LB_w}^{\varsigma}$ on the capacity sfficiency of any DRF coalition $\varsigma$ using the equations above. Furthermore, for any policy subspace $\Pi_I^E$, it is possible to compute the upper and lower bounds on the payoff of the best DRF policy in that subspace. Let $Max_s^E$ and $Min_s^E$ be the maximum and minimum (or average) payoffs of all DRF clusters in policy subspaces of size $s$. The maximum and minimum payoffs is used to bound each policy subspace for joint offloading at every level $s$, $\Pi_I^E : I \in G^n$ without necessarilly inspectig the policies they contain.

***Definition 4.4.2***. The average weighted-sum (QoS) payoff bound, and the lower weighted-sum payoff bound are equivalent.

For every logical (joint) DRF policy subspace (or Integer partition) $\Pi_I \in G^n$, let $\Pi_I\ (i)$ be the multiplicity of $i$ in $\Pi_I$ . Then it follows that:

$$\sum_{\Pi\in\Pi_I^E}\Gamma(\Pi)/|\Pi_I^E| = \sum_{i\in I}\Pi_I\,(i).\,Avg_i^E \tag{14}$$

#### 4.4.2.2 Partial search of Joint DRF policies (and ratio bound $\boldsymbol{\beta}$ from Pareto-optimal)

Solving the combinatorial problem is not a trivial task, the direct search methodology applies distributed node offloading strategy feedback to reduce the search space (ratio bound) by pruning logical Joint DRF policy subspaces using definition 4.1.

***Definition 4.4.3.*** The solution accuracy or ratio (search) bound $\beta$ from the optimal joint policy.

Consider at the federated edge, an ICG with congestion environment state $Q^-$ or free rider problem $Q^+$ for IoT task offloading and a policy subspace $\Pi^S \subseteq \Pi^E$, a ratio bound can be established on $max_{\Pi\in\Pi^E}\Gamma(\Pi)/max_{\Pi\in\Pi^S}\Gamma(\Pi)$ if and only if:

$$\forall\varsigma \subseteq E, \exists\Pi' \in \Pi^S: \sum_{i=1}^{m}\alpha_i.\,f_i\,(\gamma(\varsigma,\Pi')) = \max_{\Pi\in\Pi^E}\sum_{i=1}^{m}\alpha_i.\,f_i(\gamma(\varsigma,\Pi)) \tag{15}$$

$$\max_{\Pi\in\Pi^E}\Gamma(\Pi)\Big/\max_{\Pi\in\Pi^S}\Gamma(\Pi) \le \max_{\Pi\in\Pi^E}\left(\min_{X\in\Pi^{\Pi}:X\subseteq\,\delta^{max}(\Pi^S)}|X|\right) \tag{16}$$

In this definition, we evaluate $\delta^{max}(\Pi^S)$ as,

$$\bigcup_{\Pi'\in\Pi^S}\left\{X \subseteq \Pi': \sum_{i=1}^{m}\alpha_i.\,f_i\,(\gamma(X,\Pi')) = \max_{\Pi\in\Pi^E}\sum_{i=1}^{m}\alpha_i.\,f_i\,(\gamma(\varsigma,\Pi))\right\}.$$

***Definition 4.4.4.*** The partial search of Joint DRF policy (or Strategy profile) subspaces (worst-case bound $\beta$) is necessary for dimension reduction.

To establish a ratio bound $\beta$ on the allocation (governing) policy given an ICG with the free rider problem $Q^+$ - positive only zero-sum bargain externalities in co-offloading, every policy subspace $\Pi_I^E: I \in G^n: |I| \le 2$ must be searched. In this case, the number of Joint DRF policies searched is $2^{n-1}$ and the bound (from the optimal) $\beta = n$. Conversely, given an ICG with the congestion problem in environment states $Q^-$ (negative zero-sum bargain externalities), every subspace $\Pi_I^E: I \in \{[n],[n-1,1],[n-2,1,1],\dots,[1,1,\dots,1]\}$ must be searched, with this search, the with this search, the number of strategy profiles searched is $2^n - n + 1$ and $\beta = \left\lceil \frac{n}{2} \right\rceil$.

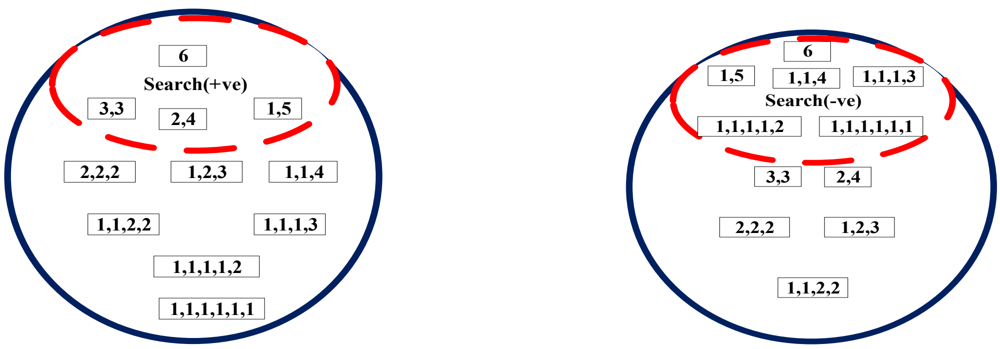
Given an ***ICG***, it will suffice to explore the first and second levels [165] of a coalition structure graph (CSG) to establish the ratio bound $\beta$. However, given the scheduling problem, it is necessary to search$\Pi_I^E: I \in \mathrm{G}^n: |I| \leq 2$ and $\Pi_I^E: I \in \{[n], [n-1,1], [n-2,1,1], \ldots, [1,1,\ldots,1]\}$ respectively (see Figure 4-5). Essentially, the best response policy or (bargain) strategy profile denoted as $\boldsymbol{\Gamma}^*$is computed as:

$$\boldsymbol{\Gamma}^* = \max_{\boldsymbol{\Pi} \in \mathbf{G}^{\mathbf{n}}} \boldsymbol{\Gamma}(\boldsymbol{\Pi}) \tag{17}$$

Distributed problem space reduction (using weighted-sum payoff bounds)

***Definition 4.4.5***. logically distributed policy subspace pruning based on the Joint DRF policy subspace graph.

Given the joint DRF policy graph of $s$ (i.e., $G^s, let\ (G^s)^{s'}$denotes the part of $(G^s)$ in which every policy subspace (node) contains at least $s'$that are 1 and where $s' < s$. Then, if we




(a) Resource non-monotonicity (free-rider states) $Q^+$ (6 nodes) (b) Population non-monotonicity (congestion states) $Q^-$ (6 nodes)

Figure 4-5: Joint DRF policy subspace partial search remove $s'$parts (that are 1) from every node in $(G^s)^{s'}$ then $(G^s)^{s'}$is like $G^{s-s'}$ as shown in A.2-**1**.

**Definition 4.4.6.** Joint DRF policy pruning - using upper and lower weighted-sum payoff bounds of Sub-policies.

Consider an ICG with congestion environment states $Q^-$ or free rider setting $Q^+$. Given a DRF cluster $\varsigma \subseteq \hat{\varsigma}$ and a sub-policy $\mathrm{X} \in \Pi^{\varsigma}$, any policy or joint DRF policy containing X can be pruned from the search space if there exist another sub-policy $\mathrm{X}' \in \Pi^{\varsigma}$ such that:

$$\forall \chi' \in \mathrm{X}', \exists \chi \in \mathrm{X}:\ \chi' \subseteq (\supseteq)\chi \ \ and\ {UB_w}^{\mathrm{X}} \leq {LB_w}^{\mathrm{X}'}$$

From definition 4.5, the following holds forth:

**Definition 4.4.7.** Joint DRF policy subspace pruning using upper and lower weighted-sum payoff bounds of Sub-policy spaces.

*Consider an ICG with congestion environment states $Q^-$ (or free rider setting $Q^+$), Then given a DRF sub-policy space $I \in G^s: s \leq n$, any search subspace represented by Joint DRF policy subspace $J \in G^n: I \subseteq J$ can be* ***pruned*** *from the search space (see Appendix A-2) if $\exists$ sub-policy space $I' \in I^s$ such that:*

$$\forall i \in I(I'), \exists J \subseteq I'(I): \textstyle\sum_{j \in J} = i\ \ and\ UB_I^W \leq LB_I^W$$

| Algorithm 5-1: The Weighted-sum Joint DRF policy Space Pruning |
|---|

1: $\hat{P} \leftarrow [\{1\}]$

2: For $s = 2\ to\ n$ do

3: {For $I \in \hat{P}$ do $\{I \leftarrow I \uplus [1]\}$}

{gets the integer partitions of $s$ without 1's}

4: $\hat{P} \leftarrow \hat{P} \cup getintegerParts(s, 2)$

5: For $I \in \hat{P}$ do {

6: If ($\boldsymbol{Q^-}$ and $\exists I' \in \hat{P}: I \rightarrow I', \boldsymbol{UB_I^W} \leq \boldsymbol{LB_{I'}^W}$)

Or ($\boldsymbol{Q}^{+}$ and $\exists I' \in \hat{P}: I' \rightarrow I, \boldsymbol{UB}_{I}^{W} \leq \boldsymbol{LB}_{I'}^{W}$) ) then $\hat{P} \leftarrow \hat{P} \backslash I$ }}

7: Return $\hat{P}$

---

### 4.4.3 Pseudo-code for hybrid scheduling policy control

Given an ICG with positive and negative controller-to-controller distributive bargain externalities, the best response policy (or *strategy profile)* $\boldsymbol{\Pi}^{*}$ is to guarantee that the fair policy payoff $\boldsymbol{\Gamma}^{*}$ is maximum or the ratio bound $\beta = 1$, either of which can be achieved via joint policy space search (see section 3.3). The IP-DRF scheduling scheme, a straightforward generalisation of the IP control algorithm (**see A.2**), is essentially a decrease and conquer procedure which comprises 2 main steps described as the process flow (Figure 4-6).

**Step I: Partial Joint DRF policy search to establish fault tolerance bounds.** Let $\mathrm{L_s}$ represent the list of DRF clusters with $\mathrm{s}$ $(\mathrm{s} \leq \mathrm{n})$ data centre agent(s) in the ICG. This stage of the algorithm computes the maximum and average payoff bounds (weighted sum) of all lists containing DRF clusters of size $\mathrm{s} \leq \mathrm{n}$. This is done by searching policy subspaces relevant to the federated edge (market failure) states (i.e., ICG with strictly negative or positive zero-sum bargain externalities). For ICGs with positive zero-sum bargaining, every policy subspace $\Pi_{\mathrm{I}}^{\mathrm{E}}: \mathrm{I} \in \mathrm{G}^{\mathrm{n}}: |\mathrm{I}| \leq 2$ must be searched whereas with negative zero-sum bargaining every subspace $\Pi_{\mathrm{I}}^{\mathrm{E}}: \mathrm{I} \in \{[\mathrm{n}], [\mathrm{n}-1,1], [\mathrm{n}-2,1,1], \ldots, [1,1,\ldots,1]\}$ is to be explored.

**(First Pass) Joint DRF policy Pruning (reduce problem complexity) using (or partial search).** In this sub procedure, the MARL scheme prunes the remaining Joint DRF policy subspaces using the (weighted sum) lower payoff bound of the most optimal Joint DRF policy and upper bargain payoff (weighted sum) bound of the rest of the subspaces (Algorithm 5-1).

**Step II: Further prune and search for Second-best Pareto optimality.** Each joint policy subspace is explored unless it is guaranteed not to contain a joint DRF policy better than the

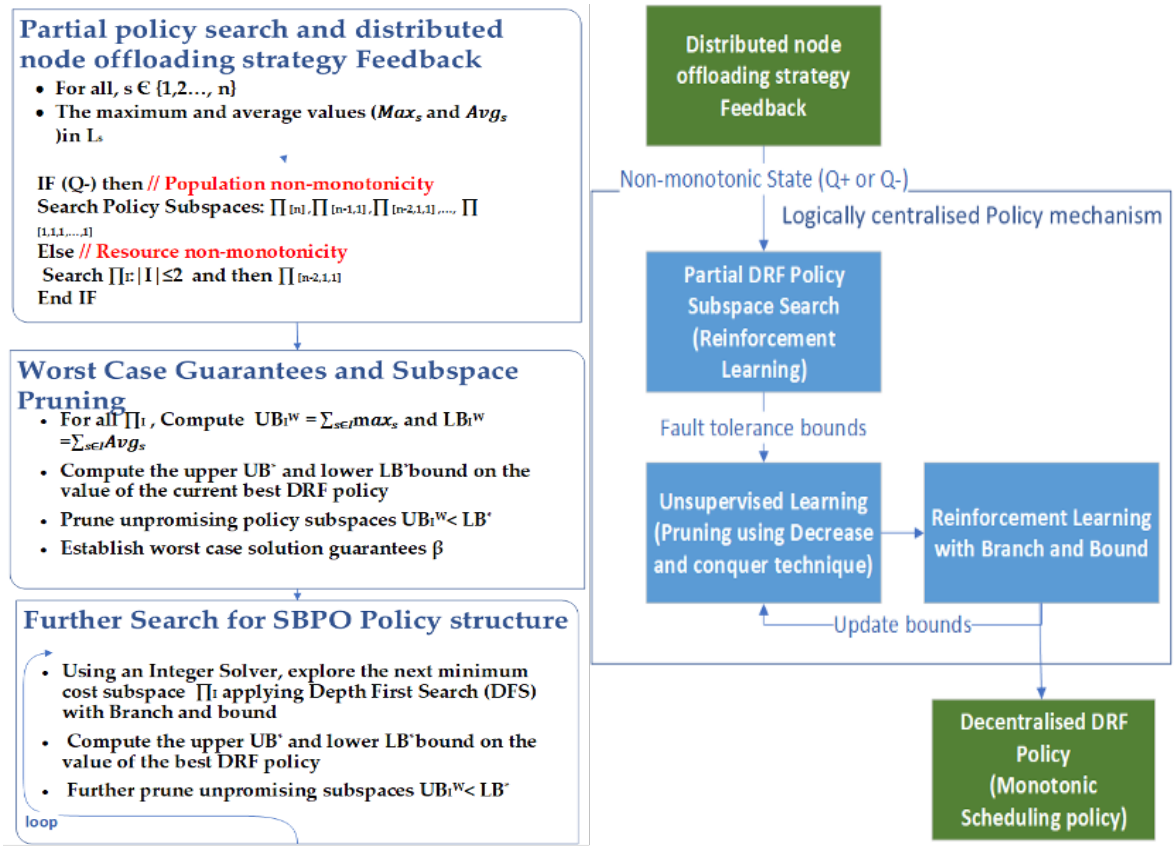



Figure 4-6: IP-DRF hybrid scheduling scheme (MARL policy)

most optimal Joint DRF policy. An integer solver [133] is used called to determine the sequence (order) in which joint DRF policies are explored based on the cost of searching policy subspaces (i.e., the number of strategy profiles in it) and current bound $\beta$. The depth first search (DFS) and branch and bound (BB) search algorithm proposed in [129], [158], [133] is applied.

## 4.5 Summary

Multi-controller SDN of the federated edge operate in two distinct sides (i.e., EC providers and consumer offloading) each with a unique user role. Rejuvenation of decentralised controller deployment (IP-DRF) must address both fault-tolerance and complexity (or behaviour) of the flat and distributed SDN control plane. The IP-DRF governs against “ageing” or potentially harmful negative and positive Controller-to-controller bargain externalities causing population and resource non-monotonicity. An outline of the research study discussed in this chapter:

- ✓ A hybrid scheduling scheme for joint and efficient computation offloading at the SDN-based (flat control plane) federated edge to due to concurrent integrative and distributive EC negotiation strategies.
- ✓ Joint DRF policy control learning framework for monotonic scheduling developed as logically centralised policy control mechanism based on physically distributed node offloading strategy feedback.
- ✓ IP-DRF scheme is a ***generalisation*** of both Integer Partition control algorithm and DRF (and DRFH) scheduling algorithms to guarantee population and resource monotonic scheduling as well as policies that are second-best Pareto-optimal (SBPO).
- ✓ The rejuvenation of decentralised controller deployment for joint offloading is inherently fast (from pruning) and returns good (bounded) solution when stopped and finally, allocations (or strategy profiles) are second-best Pareto optimal once all the promising ones have been examined.

# Chapter 5 Bottleneck fair scheduling scheme for joint forwarding in Wireless Hierarchical Networks

- This chapter introduces the rejuvenation of decentralised controller deployment in hierarchical fogs based on the Dfog-DRF scheme that applies MARL to implement a hybrid scheduling policy control mechanism for bottleneck fairness and cross-layer throughput maximisation.
- To implement the hybrid scheduling policy mechanism for joint forwarding at the wireless hierarchical (and distributed) SDN control plane. This chapter focusses on: (1) the development of a proactive reference architecture that consists of feature extraction applying multi-controller forwarding strategy feedback (homeostasis) to implement a 2-step process that evolves bottleneck (pairwise) fairness constraints (2) A stochastic game model for joint offloading applying the CCF game framework and (3) an applied orthonormal transformation framework using map reduce paradigm to sort and organise the joint policy space and (3) an application of fast MARL to determine efficient fog gateway and low-level controller's placement that ensure both load balancing for performance isolation, bottleneck fairness and group strategy proofness. Dfog-DRF aims to address cross-layer throughput maximisation of fog gateway and low-level controller's placement in joint multi-resource and multitask forwarding at the wireless hierarchical network.

## 5.1 Introduction

In the realisation of smart cities, it is imperative for backbone wireless networks (WMNs) that constitute underlying infrastructure for highly adaptive software defined networks (SDNs) to respond to changing demand in a short period of time. Run-time optimisation plays an essential role in the throughput maximisation of Fog network systems considering the service life cycle. In addition to wireless communication link requirements, specific end devices or mobiles and the compute tasks these devices offload to edge resources keep changing during the operation of the distributed fog network system. As a result, much of the operation of the fog network needs to be optimised during run-time. In light of this, the time required for executing a network optimisation algorithm is limited, therefore adopted solutions for resource allocation have to be fast [3] and fair [162] [176]. This includes the devices that make up the fog network on one hand and technical aspects such as wireless networking, computation, volatile memory on the other. An important resource allocation challenge of the wireless fog-mesh network is that several distributed wireless fog nodes and multiple resource types need to be optimised at the same time to achieve bottleneck fairness [6] while satisfying multiple service objectives e.g., latency, network usage.

Wireless hierarchical fog networks for smart cities [177] can have a relatively dynamic topology. One prominent issue in the run-time management/allocation of Wireless hierarchical fog networks is gateway and low-level controller's placement, as gateways constitute IoT traffic sinks and sources to fog networks [178]. The main objective is to maximise throughput in fog gateway and low-level controller's placement (joint) policy optimisation subject to QoS constraints of delay, gateway capacity and relay load [107] [179].

**Bottleneck fairness.** Resource management and operation of wireless SDNs means fast and efficient connection of IoT gateways to lower-level fog infrastructure is critical [179] [180]. It has been postulated in wireless fog networks [102], [181] that the network scales better when the IoT traffic pattern is localised; every node transmits only to nearby gateways within a fixed radius or distance, independent of the network size. Hence, for high throughput in fog gateway and low-level controller's placement optimisation, the wireless hierarchical fog network should be strategically split into distributed clusters, covering all IoT mesh nodes in the network graph. Time division multi-access (TDMA) link scheduling is to assign each link a set of time slots in which it can transmit in the scheduling period. Managing heterogenous real-time IoT applications in a multi-hop fog network is even more challenging. The real issue for cross-layer throughput maximisation is reduced to the physical interference on intermediate wireless (co-) links.

**Cross-Layer Throughput maximisation.** A central problem for fog gateway and low-level controller's placement optimisation of Wireless hierarchical networks for smart cities is scalability as frequently seen in multi-hop networks. To ensure weighted max-min fairness in placement of gateways considering QoS constraints, the Hierarchical WMN must grow (scale) cost-effectively and efficiently. However, as more IoT nodes are deployed and as more users participate the benefits of multi-hop wireless networks begin to fail and would degrade performance. The denser the IoT users in disjoint clusters, the more physical link interference is experienced that would deteriorate throughput or performance and scalability objectives [161].

The simple or straightforward fair scheduling scheme for throughput optimisation in fog gateway and low-level controller's placement is asset fairness [161] which tries to calculate all possible solutions and compute the optimal policy from multiple service metrics or objectives. They are mostly used for small scheduling problems since the number of possible (discrete) states of the wireless fog network increases exponentially with the number of QoS constraint dimensions.

To solve the dynamic problem, this work proposes a hybrid scheduling scheme, a mechanism design that generalises weighted DRF [6] and applies reverse game theory to ensure sharing incentive, group strategy proofness, envy-freeness and pareto-optimality; deemed essential fairness properties [176] for distributed fog environments. Applying node forwarding strategy feedback, we propose a co-channel interference model using implicit positive and negative feedback from distributed fog node monitoring to trade-off between complexity or solve time and exactness in throughput performance (maximisation) of joint gateway and low-level controller's placement. In this chapter, we propose the ***rejuvenation of decentralised controller deployment*** comprising a weighted max-min fair allocation methodology to address bottleneck fairness in distributed wireless fog micro data centre environments via generalisation of a recursive algorithm [134] to enable multi-objective (lexicographic) optimisation for cross-layer throughput maximisation.

**DFog-DRF Scheme**. The adaptive (proactive) scheme promoting the rejuvenation of decentralised controller deployment comprises a hybrid scheduling policy control mechanism based on stochastic games to address the dynamics in (multitask) co-forwarding via constraint dimensionality reduction. Implicit feedback from agent-based (distributed) monitoring maps QoS constraints for hierarchical WMNs (relay load, cluster size and hop distance) to cooperative (positive) and competitive (negative) co-channel constraints [134].

**Contribution**. To develop and implement the constraint dimensionality reduction (CDR) methodology, we propose an interference-aware model (truthful) using propositional logic to select fairness constraints for gateway and low-level controller's placement. The chapter implements a markov (repeated) game model that applies control principles of positive and negative (dual) feedback [182] to evolve (strategic) DRF constraints. Using unsupervised learning, the policy control mechanism of Dfog-DRF scheme employs a 5-stage Map-Reduce [134] framework introduced to transform positive and negative or pairwise fairness constraints resulting in isometric constraints' sets (sorted). The ordered lists foster the generation and search of feasible fog gateway and low-level controller's placement (joint) policies or strategy profiles. Finally, the DFog-DRF applies lexicographic ordering method to solve the Markov decision problem and adapt improvements in cross-layer throughput for the fog gateway and low-level controller's placement (joint) policy.

## 5.2 Reference architecture (Proactive)

**Cross-Layer throughput model.** Fog computing can be represented by a hierarchical three-layer model where the higher layers represent higher computational capacity i.e., Cloud. The middle layer (edge resources) which offers compute services near the end of the network and lowest layer comprising IoT devices (mobiles) is the focus of our interference model in muti-task forwarding.

***Definition 5-1. Formal model for Co-optimisation in the Hierarchical SDN control plane*** (see Figure 5-1). Given a set $\mathrm{E} = \{\mathrm{e}_1, \dots, \mathrm{e}_\mathrm{n}\}$ , the ***physically distributed node set*** of Mesh routers (MRs) or edge resource agents for joint computation forwarding.

$$D_M = \bigcup_{e \in E}(d_e) \tag{1}$$

$$W^E = \{\varsigma e_g : e_g \in E\} \cup \{e_g X_e : X_e, e_g \in E, X_e \nsupseteq e_g, \ |e_g| = 1\} \cup \{d_n e_n : d_n \in D_M, e_n \in \tag{2}$$

The set of all connected mobile devices $D_M$ - $d_i$ is the set of all end user devices called Mesh clients (for computation forwarding) connected to edge resource $e_i$. User mobile device $\mathfrak{d} \in d_i$ is associated with sensors $\mathfrak{s}_\mathfrak{d}$ and actuators $\mathfrak{a}_\mathfrak{d}$. Each smart edge resource $e_i$ is associated with a compute capacity $cap(e_i) \in \mathbb{R}^+$and speed $sp(e_i) \in \mathbb{R}^+$. The set of dynamic wireless (fog) communication links denoted $W^E$ includes: (1) link sets comprising links from the selected gateways $e_g$ (edge resources) to the cloud data centre $\varsigma$ (2) DRF clusters $e_g X_e$ between selected gateways and other smart routers of edge resources and (3) link set $(d_n e_n)$ for forwarding between mobile devices and edge resources. Each link $l \in W^E$ is associated with a latency $t(l) \in \mathbb{R}^+$ and a bandwidth $b(l) \in \mathbb{R}^+$. Each individual wireless link $l \in (d_n e_n)$ is associated with a relay load $ld(l) \in \mathbb{Z}^+$- the connected mobiles. The set of all fog-cloud **network resources** for forwarding is denoted as $R = \{\varsigma \cup E\}$.

**Channel Assignment**. The Hierarchical WMN is represented as a resource graph $\mathrm{G(E, \aleph)}$. Each wireless edge resource $\mathrm{e_i} = \{\mathrm{x_i}, \mathrm{y_i}, \mathrm{r_i}\} \in \mathrm{E}$ represents a smart router, where $\mathrm{x_i}$ and $\mathrm{y_i}$ are the x-coordinates and y-coordinates of $e_i$ and $r_i$ is the circular transmission range radius of $e_i$. Arc $\{e_i, e_j\} \in \aleph$ if and only if node $e_j$ is in the transmission range of router node $e_i$ or $\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \leq r_i$. However, this is not commutative as the radius of their transmission range may differ.

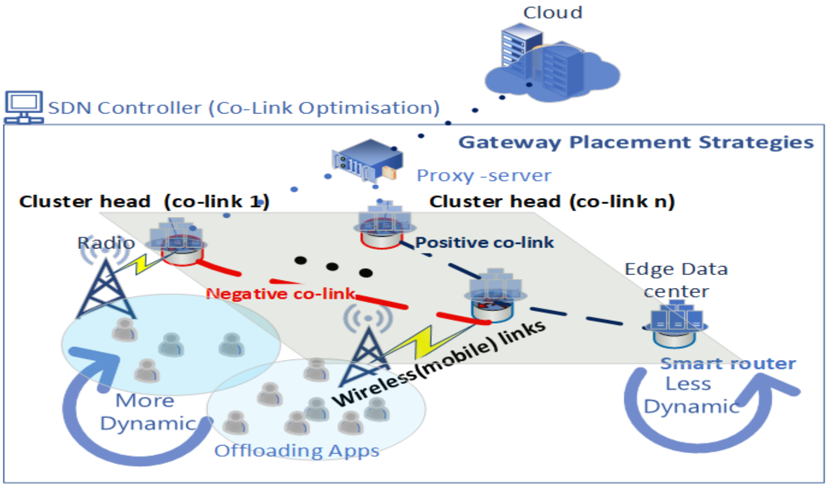



Figure 5-1: Reference architecture (bottleneck fairness) for Hierarchical SDN control plane.

**Logically distributed DRF clusters.** Each hierarchically organised fog DRF cluster is defined as a set of cooperative edge nodes $C \subseteq E$ including a gateway $e_g$ or cluster head $h_C \in C$. The edge nodes in C and the arcs between them define a DRF cluster graph (sub graph) $G_C = (C, \aleph_C)$, where an arc $\{e_i, e_j\} \in \aleph_C$ if and only if $\forall\, e_i, e_j \in C$. The maximal shortest distance between from the gateway h and all other router (agent) nodes in C is referred to as the radius of a mesh link cluster $r_C$ [108].

**Definition 5.2.** hybrid scheduling game for joint forwarding considering bottleneck fairness and throughput maximisation – A generic model of the markov game for muti-task forwarding. The muti-task forwarding (packet forwarding) markov game is a tuple $\mathcal{G} = \langle E, \widehat{CS}, V \rangle$. Let $\Pi(E)$ denote the set of all possible **logically distributed DRF clusters** that make up fog gateway and low-level controller's placement policies [45] or strategy profiles. $\widehat{CS} \subseteq \Pi(E)$ is the set of all feasible fog gateway and low-level controller's placement strategy profiles or ***joint DRF policies***. Formally, the markov game $\mathcal{G} = \langle E, \widehat{CS}, V \rangle$ is locally constrained if there exists a set of mesh DRF clusters $\mathbb{C} \subseteq 2^{E}$ such that $\widehat{CS} = \{CS \in \Pi\ (E)\ | CS \subseteq \mathbb{C}\}$, $\mathbb{C}$ is the set of feasible logic DRF clusters and V is the multi-objective value function of joint gateway and low-level controller's placement policy.

### 5.2.1 Interference-aware link scheduling model

**Definition 5.3.** Propositional logic-based feature selection for joint gateway and low-level controller's placement. The markov game for fog gateway and low-level controller's placement can be said to be propositionally definable if for feasible strategy profiles there is a gateway and low-level controller's placement feature selection constraint Φ defined over $B_E$ such that:

$$\widehat{\boldsymbol{CS}} = \{\mathrm{CS}: \mathrm{CS} \vDash \Phi\} \tag{3}$$

**Definition 5.4.** The fog gateway and low-level controller's placement feature selection method Φ can be defined as,

$$\Phi = \vee_{C \in \hat{C}} \left( (\wedge_{e_j \in C} \beta_j) \wedge (\wedge_{e_j \in E \setminus C} \neg \beta_j) \right) \tag{4}$$

On the other hand, the set of feasible fog gateway and low-level controller's placements in fog deployment be can defined as $\hat{C}$ as $\hat{C} = \{C: C \vDash \Phi\}$ given the feature selection constraint Φ.

**Markov game for bottleneck fairness considering adversarial controllers**. The muti-task forwarding game can be specified by (1) Positive DRF cluster constraints $\mathcal{P} \subseteq 2^{E}$ such that a DRF cluster C satisfies a feature selection constraint $P \in \mathcal{P}$ $(P \subseteq C)$ for fog gateway and low-level controller's placement (2) Negative co-forwarding constraints $\mathcal{N} \subseteq 2^{E}$ such that cluster C satisfies co-forwarding constraint $N \in \mathcal{N}$. $N \nsubseteq C$;

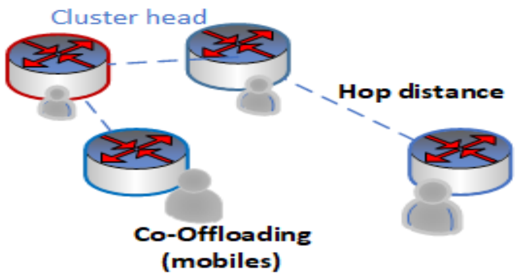



Figure 5-2: Wireless hierarchical network (co-channel) for distributed applications

**Interpretation.** The QoS constraints for wireless DRF clusters can be combined using the feature selection logic over $B_E$ (***in section 3.3.2)*** for fog gateway and low-level controller's placement, associating every co-forwarding constraint $P \in \mathcal{P}$ with $\Phi_P = \bigwedge_{e_i \in P} \beta_i$, and every $N \in \mathcal{N}$ with $\Phi_N = \neg(\bigwedge_{e_i \in N} \beta_i)$. Interpreted conjunctively, each logical DRF cluster should satisfy each negative co-forwarding constraint. Each DRF cluster should satisfy at least one positive co-forwarding constraint.

Thus, a strategy profile for fog gateway and low-level controller's placement (Joint DRF policy) $CS$ is feasible if $CS \vDash \Phi$ and where the feature selection logic is defined as,

$$\Phi_{FS} = (\vee_{P \in \mathcal{P}} \Phi_P) \wedge (\wedge_{N \in \mathcal{N}} \Phi_N) \quad (5)$$

## 5.2.2 Homeostasis: Controller forwarding strategy feedback

In this sub section, the distributed controller forwarding strategy feedback process (see Algorithm 5-1) is described under QoS constraints for wireless networks and performs the feature selection (i.e., positive, and negative fairness constraints) necessary for the QoS constraint dimensionality reduction methodology.

**QoS Constraints for Hierarchical WMN**. Given a Hierarchical WMN represented by a directed graph $G(E, \aleph)$, a delay constraint $\mathbf{R}_{\mathbf{max}}$, a relay load constraint $\mathbf{L}_{\mathbf{max}}$ and a gateway capacity constraint $\mathbf{S}_{\mathbf{max}}$, the real-time fog gateway and low-level controller's placement problem is to find a set of DRF clusters $\{C_1, C_2, \ldots, C_m\}$ and their representative minimum Steiner flow trees such that n is minimal subject to:

a) $C_1 \cup C_2 \cup \ldots \cup C_m = E$

b) $|C_k| \leq S_{max}$, where $1 \leq k \leq n$.

c) $r(C_k) \leq R_{max}$, where $1 \leq k \leq n$.

d) $\forall e \in T(G_{C_K}), \psi(e) \leq L_{max}$.

**Hop distance constraint ($R_{max}$) as Wireless DRF cluster delay.** Real-time application delay constraint is translated into an upper bound R on the mesh cluster radius. The Fog-based Steiner (rectilinear) tree is a shortest-path spanning tree of $G_C$, $T(G_C)$ (see Figure 5-2) established by obtaining the shortest paths from the gateway h to all other router link nodes $X_e$ in C. All router nodes at ith level of the minimum Steiner tree are i hops distant to the gateway or link cluster head h. The quantity dep (T ($G_C$)) denotes the depth of $T(G_C)$. With a controller node e in $T(G_C)$, the number of nodes in the subtree rooted e is denoted $\psi$(e).

The mobility of router nodes can lead to new cooperative arrangements or renders some co-forwarding strategies infeasible due to the maximum hop distance Rmax. Within mesh DRF clusters, Rmax guarantees a bound on the maximum expected latency or delay of real-time applications,

$$r(c_i, \zeta) \leq R_{max}\ , \forall \zeta \in C_i \text{ and } \zeta \neq c_i \tag{6}$$

$r(c_i, \zeta)$ is the depth of Steiner tree in terms of number of hops (distance), between gateway $c_i$ and transmission node $\zeta$.

**DRF cluster size constraint ($S_{max}$).** For multiple resources, a threshold is put on the number of fog node links that can be supported by each mesh cluster and the gateway to provide guarantees for real-time applications. A minimum throughput for each cluster minimises the blocking (interference) probability between a node and its gateway.

$$|C_i| \leq S_{max}\ \forall C_i \subseteq E \tag{7}$$

**DRF cluster (relay) load constraint ($L_{max}$).** User mobility in mesh-IoT clusters may lead to new cooperative and competitive strategies for real-time applications due to dynamic load of the intermediate node links, and it can be expressed as,

$ld(\mathrm{v}) \leq \mathrm{L}_{max}$ (8)

### 5.2.3 Pairwise-fairness constraints for homeostatic regulation

A feasible fog gateway and low-level controller's placement (joint) policy is generated from the positive and negative co-forwarding constraint set modelled as dynamics of the strategic markov game. Algorithm 5-1 outline the pseudo-code to compute $\Phi_{FS}$ using feedback from monitoring (smart) router agents.

*Example 5.1. Muti-task forwarding (channel allocation) game in Wireless Fog-Mesh network (Positive and Negative co-forwarding or bottleneck fairness constraints in* ***Figure 5-3****) to represent the ageing of multi-controller SDN.* Consider a markov game with seven fog resource nodes (**MRs**) given as $\mathrm{E} = \{\mathrm{e}_1, \mathrm{e}_2, \mathrm{e}_3, \ldots, \mathrm{e}_7\}$ and **MCs** denoted $\mathrm{D}_e = \{\mathrm{d}_1, \mathrm{d}_2, \mathrm{d}_3, \ldots, \mathrm{d}_7\}$. Assume resource nodes can form DRF clusters to meet the QoS requirements of IoT applications in the wireless (hierarchical) mesh network. While each edge resource or **smart MR** has a processing capacity and bandwidth for *application module placement and data transmission*, application links $l \in W^E$ is associated with a latency $t(l) \in \mathbb{R}^+$ and a bandwidth $b(l) \in \mathbb{R}^+$. Each individual wireless link $l \in (d_n e_n)$ is associated with a relay load $ld(l) \in \mathbb{Z}^+$- the connected mobiles and resource requirements.

**Step 1: hierarchical organisation – master and low-level controller nodes.** Prior knowledge (online monitoring/feedback) shows that certain hierarchical node sub-clusters in co-forwarding are minimal and indispensable to satisfy the QoS constraint of hop distance i.e., $r(e_g, e_j) \ll \boldsymbol{R_{max}} \in \mathbb{Z}^+ \forall\, e_j \in X_e$. Assume at some discrete time t, let $\mathcal{M}_t^{\Phi}$ denote the set of Steiner subsets satisfying $\boldsymbol{R_{max}}$. Let's say this set of structured elements is given by $\mathcal{M}_t^{\Phi}$ $=\{\{d_{157}e_{157}\}, \{d_{256}e_{256},\}, \{d_{567}e_{567}\}, \{d_{123}e_{123}\}, \{d_{235}e_{235}\}\}$.

**Step 2: interference-aware constraints**. At discrete time t, let edge resource node alliances $\mathcal{P}_t$ or constraints (Steiner resource [183] subtrees) $\mathrm{P}^1{}_\mathrm{t} = \{d_{157}e_{157}\}$, $\mathrm{P}^2{}_\mathrm{t} = \{d_{256}, e_{256}\}$ and $\mathrm{P}^3{}_\mathrm{t} = \{d_{567}e_{567}\}$ reliably satisfy QoS constraints of link relay workload $ld(\mathrm{P}_1), ld(\mathrm{P}_2), ld(\mathrm{P}_3) \ll \boldsymbol{L_{max}} \in \mathbb{Z}^+$ and size constraint $|\mathrm{P}_1|, |\mathrm{P}_2|, |\mathrm{P}_3| \ll \boldsymbol{S_{max}} \in \mathbb{Z}^+$ of computation offloading requirements.

On the other hand, for joint gateway and low-level controller's placement, any DRF sub-cluster $\mathcal{N}_t$ involving edge node alliances (link sub-clusters) that are negative $\mathrm{N}^1{}_\mathrm{t} = \{d_{123}e_{123}\}$, $\mathrm{N}^2{}_\mathrm{t} = \{d_{235}e_{235}\}$ should be avoided because these specific combinations violate relay link workload constraints $ld(\mathrm{N}_1), ld(\mathrm{N}_2) \geq \boldsymbol{L_{max}}$ thereby affecting QoS guarantees for some forwarding applications.

Algorithm 5-1: (Fairness Constraints) gateway placement feature selection process applying node offloading strategy feedback

**Step1**. (hierarchical constraints). Monitoring for application of the real-time delay constraint ($\mathbf{R}_{\mathbf{max}}$) to G, results in the minimum Steiner tree set $\mathcal{M}_t^{\Phi}$, $\mathcal{M}_t^{\Phi}$= subsets$(\mathrm{G}, \mathbf{R}_{\boldsymbol{max}}, \mathrm{S})$ .

**Step2.** (interference-aware constraints). Applying relay load and cluster size constraints to $M_t$, positive and negative co-forwarding constraint is (ineq. 7, & 8) computed as:

- $\forall P \in \mathcal{M}_t^{\Phi}$ such that $(l(P) \ll L_{max}$ and $|P| \ll S_{max})$: $\Phi_\mathrm{P} = \bigwedge_{\mathrm{e_i} \in \mathrm{P}} \beta_\mathrm{i}$
- $\forall N \in \mathcal{M}_t^{\Phi}$ such that $(l(N) \geq L_{max}$ $or$ $|N| \geq S_{max})$: $\Phi_\mathrm{N} = \neg(\bigwedge_{\mathrm{e_i} \in \mathrm{N}} \beta_\mathrm{i})$

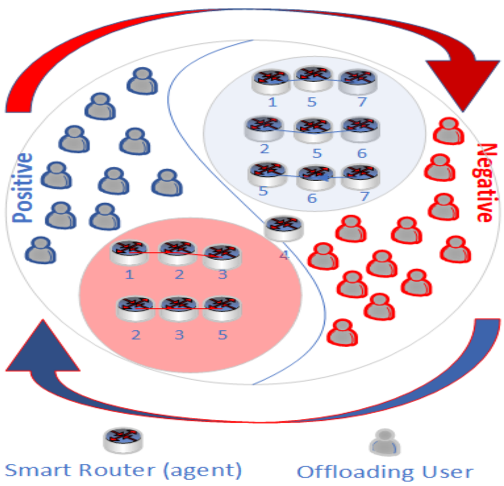



Figure 5-3: Pairwise-fairness constraints in Wireless hierarchical Fogs (Example 5.1)

## 5.3 System model

**The fog link scheduling problem**. The joint multitask and multi-resource forwarding (markov) game $\mathcal{G}$ can be defined as: $\mathcal{G} = \langle E, k, \mathcal{P}, \mathcal{N}, \mathrm{S}_{max}, \sigma, v, V \rangle$, the lexicographic or multi-objective value function, $V: 2^E \rightarrow \mathbb{R}^r$ is a and real-valued characteristic function on $E$, At discrete scheduling time $t \in \sigma$, $\mathcal{P}_t$ *and* $\mathcal{N}_t$ are sets of subsets of $E$ and $\mathrm{S}_{max}$ is the DRF cluster size constraint.

A logically distributed DRF cluster $C \subseteq E$ is feasible for $\mathcal{G} = \langle E, \mathcal{P}, \mathcal{N}, \mathrm{S}_{max}, \sigma, v, V \rangle$, if: $P \subseteq C$ for some $P \in \mathcal{P}_t$, $\neg(N \subseteq C), \forall N \in \mathcal{N}_t$ and $|C| \in \mathrm{S}_{max}$ and scheduling duration $t \in \sigma$.

Essentially, a fog DRF cluster is feasible if it contains at least one of the hierarchical positive co-forwarding constraints and none of the negative co-forwarding constraints. We denote by $\mathbb{C}\langle E, \mathcal{P}, \mathcal{N}, \mathrm{S}_{max} \rangle$ the set of all feasible DRF clusters or controller node strategies**.** Therefore, a strategy profile CS for fog gateway and low-level controller's placement is only feasible if and only if $\mathrm{CS} \subseteq \mathbb{C}\langle E, \mathcal{P}, \mathcal{N}, \mathrm{S}_{max} \rangle$. Given any feasible fog gateway and low-level controller's placement (joint) policy or strategy profile CS, we set,

$$V(CS) = \sum_{i=1}^{m} \beta_i f_i\left(\sum_{C \in CS} v(C)\right) \tag{9}$$

$\sum_{i=1}^{m} \beta_i = 1$ is the priority-based scalarisation factor, $f_i$ is the set of objectives for throughput and $\sum_{C \in CS} v(C)$ represents the resource function - aggregated routing of data, code, and tasks within each DRF cluster of the fog gateway and low-level controller's placement policy.

$$CS_t^{\,*} = \max_{CS \in \Pi(\mathrm{E})} V(CS) \tag{10}$$

With the feature selection for fairness constraints described in the Example 5.1, the problem can now be specified given the basic Muti-task forwarding (packet forwarding) game described.

Given $\boldsymbol{k}$ a set of the minimal number of fog gateways deployable (denoted by k) and the interference model in the fog network, we study optimisation of fog gateway and low-level controller's placement in the hierarchical fog network. In the proposed scheme, total throughput is maximised while also providing guarantees of bottleneck fairness among all forwarding user applications – as well as avoiding scheduler manipulation.

Applying the constrained coalition formation (CCF) game [134] framework, we model the forwarding game with dynamics that aims to investigate interference-awareness in dynamic wireless fog networks. Node forwarding strategy feedback (distributed monitoring) is used to implement fairness constraints (positive and negative) based on the pairwise gateway feature selection $\Phi_{FS}$ of the constraint dimensionality reduction methodology. To achieve this goal, we define the fog gateway and low-level controller's placement feature selection method over propositional logic and address the physical link scheduling problem for joint forwarding as dynamics in the markov decision process.

The goal is to find feasible fog gateway and low-level controller's placement (joint) policy or strategy profile $CS^*$ that maximises throughput $V(CS)$ amongst all feasible gateway policies or strategy profiles. The next section describes the MARL methodology or centralised policy control mechanism applied to solve this link scheduling problem.

Computing the fog network throughput V(CS) is implausible in severe situations using a centralised approach, especially with multiple often conflicting objectives of latency performance and resource usage. This huge combinatorial optimisation problem needs to be reduced to sub-problems that can be solved.

To overcome this challenge, we propose a more robust way of establishing weighted max-min fairness, the Dfog-DRF scheme. Our solution proposes the use of distributed node strategy feedback to generate positive and negative co-forwarding (fairness) constraints for the centralised policy control. A transformation (Map-Reduce) framework that applies divide and conquer to build isometric or principal co-forwarding constraint sets. The proposed novel forwarding scheme is Dfog-DRF that avoids searching redundant or infeasible fog gateway and low-level controller's placement policies thereby speeding up solution time (scalability) while guaranteeing cross-layer throughput-awareness via lexicographic multi-objective optimisation.

## 5.4 Robust MARL against adversaries (DFog-DRF scheme)

This sub-section presents the mechanism design of feasible fog gateway and low-level controller's placement (joint) policies using a two-step constraint dimension reduction methodology as shown in Figure 5-4. First step in the *fair allocation methodology* is to have our feasible policy control space sorted in some order. An extraction/transformation $\mathcal{F}$ process converts the specified set of fairness (from node forwarding strategy feedback) constraints to another isomorphic or congruent set $T^*$. Divide and conquer (D&C) technique proposed in [134] provides an isometric transformation method to simplify the identification of all feasible fog gateway and low-level controller's placement policies in a quick manner (i.e., avoids performing redundant computations). Second, fast joint policy control based on lexicographic multi-objective optimisation is proposed to find the optimal fog gateway and low-level controller's placement (joint) policy using a recursive algorithm.

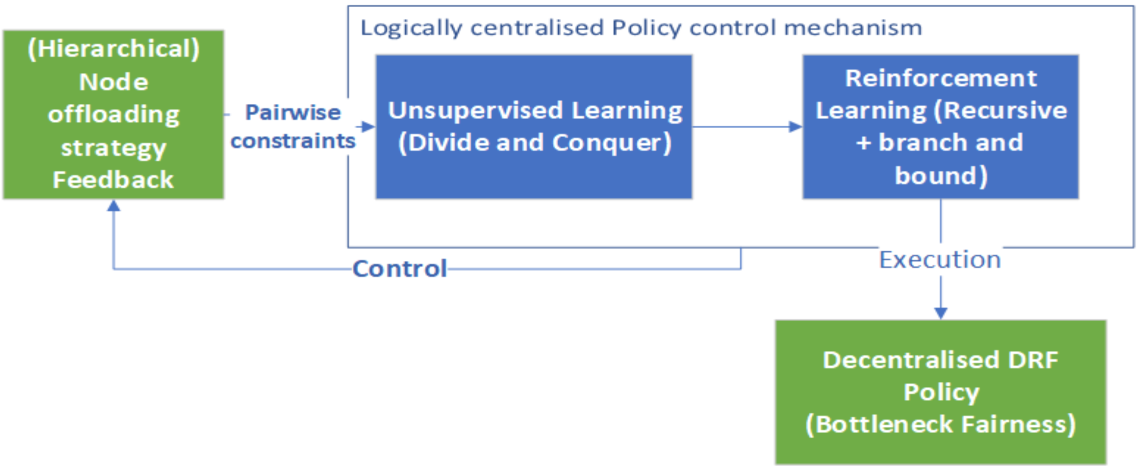



Figure 5-4: DFog-DRF scheme (MARL Policy)

### 5.4.1 Definitions

**Definition 5.1.** A base case fog gateway and low-level controller's placement (clustering) problem is where $\mathbb{C}\langle \mathrm{E}, \mathcal{P}, \mathcal{N}, \mathrm{S}_{max}\rangle$ satisfies the following conditions:

- The size of the set containing positive co-forwarding constraints is minimum and negative co-forwarding constraints is empty $|\mathcal{P}| = 1, \mathcal{N} = \emptyset$ .
- The number of negative co-forwarding constraints in the set is at most 1 $\left|\{N \in \mathcal{N} \middle| |N| > 1\}\right| \leq 1$.

**Definition 5.2.** Given any link node $e_i \in E$, with mobile device link set $d_i e_i$, the following holds forth:

$$\mathbb{C}\langle E, \mathcal{P}, \mathcal{N}\rangle = \mathbb{C}\langle E\backslash\{e_i\}, \mathcal{P}^{\bar{e}_i}, \mathcal{N}^{\bar{e}_i}\rangle \cup \langle\{e_i\}\oplus\mathbb{C}\langle E\backslash\{e_i\}, \mathcal{P}^{\bar{e}_i} \cup \widetilde{\mathcal{P}^{e_i}}, \mathcal{N}^{\bar{e}_i} \cup \widetilde{\mathcal{N}^{e_i}}\rangle\rangle \qquad (15)$$

**Definition 5.3**. *Sorted-order principal pairwise-fairness constraints – essential to speed up fog gateway and low-level controller's placement (joint) policy search.*

**Background**. We apply the divide and conquer algorithm (unsupervised learning) for distance (colinear) preservation; we are required to specify:

1. Feasible DRF clusters of $\mathbb{C}\langle \mathrm{E}, \mathcal{P}, \mathcal{N}, \mathrm{S}_{max}\rangle$.
2. Define a base case clustering problem for gateway and low-level controller's placement.
3. Specify how sub-problems are generated.

To generate feasible DRF clusters of $\mathbb{C}\langle \mathrm{E}, \mathcal{P}, \mathcal{N}, \mathrm{S}_{max}\rangle$, let's define the following variables:

$\mathcal{N}^{\wedge} = \{N \in \mathcal{N} |\ |N| > 1\}$, $\mathrm{N}^{\wedge\wedge} = \mathcal{N}\backslash\mathrm{N}^{\wedge}$, $E^{\wedge} = \cup\, \mathcal{N}^{\wedge}$ and

$E^{\wedge\wedge} = E\backslash \cup\, (\mathcal{P} \cup \mathcal{N}^{\wedge} \cup \mathcal{N}^{\wedge\wedge})$.

Given a base case clustering problem, the following holds true ($\otimes$ is cartesian product):

$$\mathbb{C}\langle \mathrm{E}, \mathcal{P}, \mathcal{N}\rangle = \mathcal{P} \otimes \bigcup_{s \in S_{max} \cup \{0\}, s < |E^{\wedge}|} \mathcal{g}_s(E^{\wedge}) \otimes \bigcup_{s \in S_{max} \cup \{0\}, s \leq |E^{\wedge\wedge}|} \mathcal{g}_s(E^{\wedge\wedge}) \qquad (11)$$

Let $\mathcal{g}_s$ be a function that takes the set of link nodes in $\mathrm{C} \subseteq \mathrm{E}$ as input and returns all sub-clusters of DRF cluster C that are of size s. This means $\mathcal{g}_s(C) = \{C^*: C^* \subseteq C, |C^*| = s\}$. Then with a base case and $\mathbb{C}\langle \mathrm{E}, \mathcal{P}, \mathcal{N}, \mathrm{S}_{max} \rangle$, this approach can generate required DRF clusters without checking fairness constraints based on this definition.

The transformation approach divides the huge problem of generating link clusters $\mathbb{C}\langle \mathrm{E}, \mathcal{P}, \mathcal{N}, \mathrm{S}_{max} \rangle$ into two sub-problems - link clusters that contain a specific node are separated from those that do not contain link node $\mathrm{e}_\mathrm{i}$. Generating DRF clusters that have controller node $\mathrm{e}_\mathrm{i}$ requires removing $\mathrm{e}_\mathrm{i}$ from every positive or negative co-forwarding constraint. Likewise, generating link clusters that do not contain $\mathrm{e}_\mathrm{i}$ suggests we remove every positive or negative co-forwarding constraint from $\mathcal{P}$ and $\mathcal{N}$ that contains edge node agent $\mathrm{e}_\mathrm{i}$. Because this operation is done recursively, keeping track of nodes that must and must not be contained in DRF clusters in the new principal co-forwarding constraints' sets $\mathcal{P}^*$ and $\mathcal{N}^*$.

### 5.4.2 Unsupervised framework

The Transformation function $\mathcal{F}$ to generate equivalent fairness constraint sets takes as input $(\mathrm{E}, \mathcal{P}, \mathcal{N})$ and returns $\mathcal{F}: (\mathrm{E}, \mathcal{P}, \mathcal{N}) \rightarrow (\mathrm{E}^* \, T^*)$, which includes an orthogonal (**linear**) transformation $T^*\langle \mathcal{P}, \mathcal{N} \rangle = \langle \mathcal{P}^*, \mathcal{N}^* \rangle$ and $E^*$ is a set used to sort DRF clusters. The main stages of the orthonormal transformation algorithm [134] for the preservation of structured fairness constraints are as follows:

**Stage 1: Decentralised control problem Initialisation**

The theoretical basis for this sub-process is formed by defining a DRF cluster sets of sets operand $\oplus$. Let Å and $\mathcal{B}$ be two arbitrary sets of sets then,

$$\text{Å} \oplus \mathcal{B} = \begin{cases} \{\mathrm{A} \cup \mathrm{B} \mid \mathrm{A} \in \text{Å}, \mathrm{B} \in \mathcal{B}\} & if \ \text{Å} \neq \emptyset, \mathcal{B} \neq \emptyset \\ \emptyset & otherwise \end{cases} \tag{12}$$

For any edge node $e_i \in E$, let $\mathcal{P}^{e_i} = \{C \in \mathcal{P} \mid e_i \in C\}$ and

For any edge node $e_i \in E$, let positive constraint set $\mathcal{P}^{e_i} = \{C \in \mathcal{P} \mid e_i \in C\}$ and $\mathcal{P}^{\overline{e_i}} = \mathcal{P} \backslash \mathcal{P}^{e_i}$, similarly, let $\mathcal{N}^{e_i} = \{C \in \mathcal{N} \mid e_i \in C\}$ and $\mathcal{N}^{\overline{e_i}} = \mathcal{N} \backslash \mathcal{N}^{e_i}$. Concluding we define $\widetilde{\mathcal{P}^{e_i}}$ and $\widetilde{\mathcal{N}^{e_i}}$ as,

$$\widetilde{\mathcal{P}^{e_i}} = \{\, C \subseteq 2^E \setminus \{e_i\} \mid C \cup \{e_i\} \in \mathcal{P}^{e_i}\} \tag{13}$$

$$\widetilde{\mathcal{N}^{e_i}} = \{\, C \subseteq 2^E \setminus \{e_i\} \,|C \cup \, \{e_i\} \, \in \, \mathcal{N}^{e_i}\} \tag{14}$$

A branching router node is first selected and stored in set $E^*$only if $\mathcal{P}^* = \emptyset$. Reinforcement learning via Joint DRF policy control (using principal constraint lists) is the reason behind the use of set $E^*$. The node at the ith location in $E^*$is $e_i^*$. This process also sets $\mathcal{P}^{\bar{e_i}}, \mathcal{N}^{\bar{e_i}}, \widetilde{\mathcal{P}^{e_i}}, \widetilde{\mathcal{N}^{e_i}}$ as they are necessary for the divide and conquer technique of the constraints-based dimensionality reduction.

**Stage 2: Problem decomposition method (D&C)**

**Application of D&C.** With definitions described above, we now provide an approach that allows us to decompose (locally) the pairwise-fairness constraints by choosing branching link node $e_i$ and dividing the problem into two smaller (sub) problems.

The controller placement problem is divided into two base problems (using Definition 5.2). The scheduling algorithm is guaranteed to reach one of the exit conditions $\mathcal{P} \ni \emptyset \wedge \mathcal{N} = \emptyset$ or $\mathcal{P} = \emptyset$, $\mathcal{N} \ni \emptyset$ because of the way $\mathcal{P}$ *and* $\mathcal{N}$ are updated. The updates involve either the removal of entire link sub-clusters or nodes. Finally, principal co-forwarding constraints $\mathcal{P}^*, N^*$ are put together in such a way that they are guaranteed to be base case distributed clustering problems. Also, an example of how the method does the orthonormal transformation is given in Figure 5-5.

Stages 1 and 2: Initialisation and D&C

// D&C and Initialisation

1: $\mathrm{e_i} \leftarrow \mathrm{choose}(\mathrm{E}, \mathcal{P}, \mathcal{N}, \mathcal{P}^*, \mathcal{N}^*)$

2: $\mathcal{N}^{\bar{e_i}} \leftarrow \emptyset$ and $\widetilde{\mathcal{N}^{e_i}} \leftarrow \emptyset, \mathcal{P}^{\bar{e_i}} \leftarrow \emptyset$ *and* $\widetilde{\mathcal{P}^{e_i}} \leftarrow \emptyset$

3: If $\mathcal{P}^* = \emptyset$ then $\mathrm{E}^* \leftarrow \mathrm{E}^* \cup \{\mathrm{e_i}\}$

********************

4: For all $\mathrm{N} \in \mathcal{N}$ do

5: If $\mathrm{e_i} \in \mathrm{N}$ then $\widetilde{\mathcal{N}^{e_i}} \leftarrow \widetilde{\mathcal{N}^{e_i}} \cup \{\mathrm{N} \setminus \{\mathrm{e_i}\}\}$

6: else If $\mathrm{e_i} \notin \mathrm{N}$ $\mathcal{N}^{\bar{e_i}} \leftarrow \mathcal{N}^{\bar{e_i}} \cup \{\mathrm{N}\}$ End if

7: End for

**********************

For all $P \in \mathcal{P}$ do

{If $e_i \in P$ then $\widetilde{\mathcal{P}^{e_i}} \leftarrow \widetilde{\mathcal{P}^{e_i}} \cup \{P \backslash \{e_i\}\}$

else If $(e_i \notin P)$ $\mathcal{P}^{\bar{e_i}} \leftarrow \mathcal{P}^{\bar{e_i}} \cup \{P\}$ End if

End for

----D&C Application for subproblems---

$\mathcal{F}(E\backslash\{e_i\}, \mathcal{P}^{\bar{e_i}} \cup \widetilde{\mathcal{P}^{e_i}}, \mathcal{N}^{\bar{e_i}} \cup \widetilde{\mathcal{N}^{e_i}}, \{\bigcup_{\boldsymbol{P}\in\boldsymbol{\mathcal{P}}^*\cup\{\boldsymbol{e_i}\}} \boldsymbol{P}\}, \boldsymbol{\mathcal{N}}^*)$

$\mathcal{F}(E\backslash\{e_i\}, \mathcal{P}^{\bar{e_i}}, \mathcal{N}^{\bar{e_i}}, \boldsymbol{\mathcal{P}}^*, \boldsymbol{\mathcal{N}}^* \cup \{\boldsymbol{e_i}\})$

**Stage 3**: Purge redundant fairness constraints. In this stage, and occurring at every call of the recursive function, it becomes essential to remove:

- Every negative fairness constraint that are supersets of another negative constraint.
- Every positive fairness constraint that is a superset of another one.

Stage 3: (Reduce) redundant fairness constraints

//Ensure non-redundant negative Co-forwarding constraints

For all $N \in \mathcal{N}$ do

If $\exists N^{\wedge} \in \mathcal{N}: N^{\wedge} \supset N$ then **remove** $\boldsymbol{N^{\wedge}}$ **from** $\boldsymbol{\mathcal{N}}$

//Ensure non-redundant positive Co-forwarding constraints

For all $P \in \mathcal{P}$ do

If $\exists P^{\wedge} \in \mathcal{P}: P^{\wedge} \supset P$ then remove $\boldsymbol{P^{\wedge}}$ **from** $\boldsymbol{\mathcal{P}}$

**Stage 4: Functional mapping.** Cases in which only one co-forwarding constraint (either positive or negative) remains to be met – the base case gateway and low-level controller's placement problem. In such cases, the orthogonally transformed positive and negative co-forwarding constraints are computed.

**Stage 5: Check for exit conditions.** Based on the definitions of $\mathcal{P}$ and $\mathcal{N}$, the positive co-forwarding constraints are met if $\mathcal{P} \ni \emptyset$ and the negative co-forwarding constraints are satisfied only if $\mathcal{N} = \emptyset$. We define a set $T^*$ which stores the tuple $(\mathcal{P}^*, \mathcal{N}^*)$ if both $\mathcal{P}$ and $\mathcal{N}$ are satisfied. The co-forwarding constraints would not be satisfied if $\mathcal{P} = \emptyset$ or $\mathcal{N} \ni \emptyset$.

---

Stages 4 and 5: Map and Exit conditions

---

1: For all $N \in \mathcal{N}$ : $|N| = 1$ do

2: $E \leftarrow E\backslash\{N\}$ ; $\mathcal{N}^* \leftarrow \mathcal{N}^* \cup N$; $\mathcal{N} \leftarrow \mathcal{N}\backslash\{N\}$

3: End For

4: If $\mathcal{P} \ni \emptyset$ and $|\mathcal{N}| = 1$ then $\mathcal{N}^* \leftarrow \mathcal{N}^* \cup \mathcal{N}$, $\mathcal{N} \leftarrow \emptyset$

5: If $|\mathcal{P}| = 1$ and $\mathcal{N} = \emptyset$ then $\mathcal{P}^* \leftarrow \bigcup_{P \in \mathcal{P} \cup \mathcal{P}^*} P$; $\mathcal{P} \leftarrow \{\emptyset\}$

---Exit Conditions---

6: If $\mathcal{P} \ni \emptyset$ and $\mathcal{N} = \emptyset$ then $T^* \leftarrow T^* \cup \{(\mathcal{P}^*, \mathcal{N}^*)\}$; Exit

7: If $\mathcal{P} = \emptyset$ and $\mathcal{N} \ni \emptyset$ then Exit

// Check Wireless DRF cluster size constraints

8: If $(|\cup \mathcal{P}^*| > Max(S_{max}))$ or $(|E| - |\mathcal{N}^*| < Min(S_{max}))$ Exit

---

### 5.4.3 Fog scheduling policy control mechanism

The goal of MARL is to find the gateway and low-level controller's placement (GP) strategy profile $CS^*$ that maximises throughput, the scalarised $V(CS)$ of all feasible cluster $\mathbb{C}\langle E, \mathcal{P}, \mathcal{N}, S_{max}\rangle$. The control method is generalised to handle multiple objectives and avoid the costly route of going through every possible combination in the scheduling problem space. The Dfog-DRF scheme makes use of the principal constraints orthonormal transformation described in the previous section. The set of all feasible logically distributed link clusters $\mathbb{C}\langle E, \mathcal{P}, \mathcal{N}, S_{max}\rangle$ is divided into principal (sorted) constraint lists $\mathcal{L}_1, \dots, \mathcal{L}_{|E^*|+1}$ Where $\mathcal{L}_j = \{(\mathcal{P}^*, \mathcal{N}^*): \mathcal{P}^* \ni e_j^*\}$, for $j = 1, \dots, |E^*|$, $\mathcal{L}_{|E^*|+1} = \{(\mathcal{P}^*, \mathcal{N}^*): \mathcal{P}^* \cap E^* = \emptyset\}$. The following definition explains the reasoning behind this division.

**Explanation (Definition 5.3).** Every feasible gateway and low-level controller's placement policy or strategy profile contains exactly one wireless DRF cluster from $\mathcal{L}_1$ and at most one wireless DRF cluster from every $\mathcal{L}_\mathrm{j}$ where j $\in \{2, \ldots, |\mathrm{E}^*| + 1\}$.

The centralised scheduling policy control mechanism of Dfog-DRF scheme for joint gateway and low-level controller's placement policy optimisation is provided in Algorithms 5-2 & 5-3 and a pictorial view of how it searches through wireless DRF cluster lists is given in Figure 5-

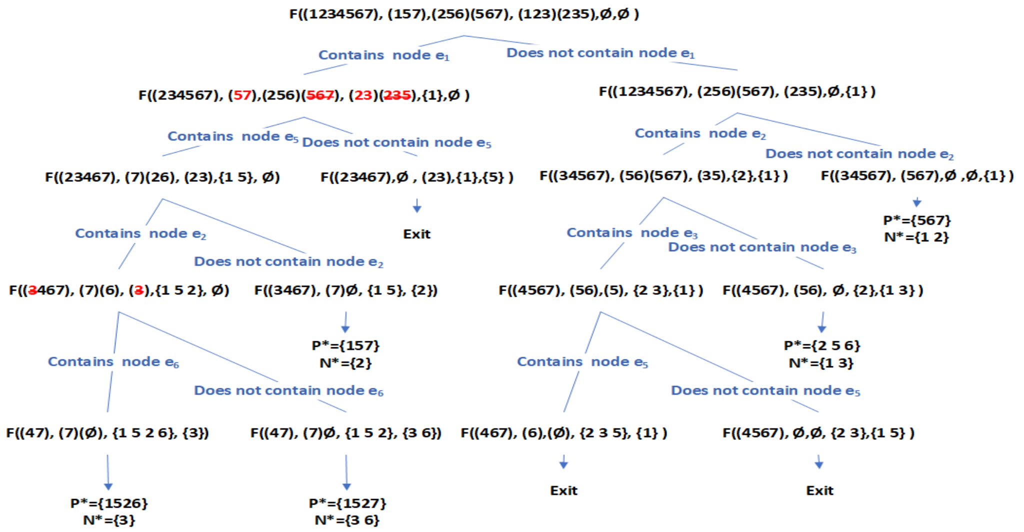



Figure 5-5: MapReduce (linear) transformation process to obtain Principal variable sets given +Ve and -Ve Co-forwarding constraints (Example 5.1)

6. Essentially, the method fills up the principal constraint lists $\mathcal{L}_1, \ldots, \mathcal{L}_{|\mathrm{E}^*|+1}$ and then calculates for DRF clusters in each $\mathcal{L}_i$ , a weighted-sum (lexicographic ordering) upper bound $\mathrm{UB}^W{}_{\mathcal{L}_\mathrm{i}}$ as shown in steps 1 to 18. In the next step, Algorithm 5-3 (a function) is called which operates by adding wireless DRF clusters to $CS$ until it is a feasible fog gateway and low-level controller's placement (joint) policy or strategy profile. After which it updates if necessary $CS^*$, the most optimal gateway and low-level controller's placement policy or strategy profile found at this point.

Algorithm 5-2: Hybrid Scheduling policy control (DFog-DRF scheme)

**Algorithm. 5-2: HybridDRFScheduler**$(\mathbf{E}^*,\ \mathbf{T}^*, \{\boldsymbol{\beta_1}, \dots, \boldsymbol{\beta_m}\}, \{\boldsymbol{f_1}, \dots \boldsymbol{f_m}\})$

**global k // set of number of deployable gateways**

For $\mathrm{j} = 1$ to $|\mathrm{E}^*|$ do

    $\mathcal{L}_\mathrm{j} \leftarrow \emptyset$; $\mathbf{UB}^{\boldsymbol{W}}{}_{\mathcal{L}_\mathrm{j}} \leftarrow -\infty$ {initialise $\mathcal{L}_\mathrm{j}$ and $\mathrm{UB}^{W}{}_{\mathcal{L}_\mathrm{j}}$ }

End for

For $(\mathcal{P}^*, \mathcal{N}^*) \in \mathrm{T}^*$ do

Listfound $\leftarrow$ false

    For $\mathrm{j} = 1$ to $|\mathrm{E}^*|$ do {if $\mathcal{P}^*$ contains $\mathrm{e}_\mathrm{j}^*$ then update $\mathcal{L}_\mathrm{j}$ and $\mathrm{UB}_{\mathcal{L}_\mathrm{j}}$ }

    If $\mathrm{e}_\mathrm{j}^* \in \mathcal{P}^*$ then $\mathcal{L}_\mathrm{j} \leftarrow \mathcal{L}_\mathrm{j} \cup (\mathcal{P}^*, \mathcal{N}^*)$

        For $\mathrm{C} \in \mathbb{C}(\mathcal{P}^*, \mathcal{N}^*)$ do

            If $\sum_{i=1}^{m} \boldsymbol{\beta_i f_i}(\boldsymbol{v}(\boldsymbol{C})) > \mathbf{UB}^{\boldsymbol{W}}{}_{\mathcal{L}_\mathrm{j}}$ then $\mathbf{UB}^{\mathbf{W}}{}_{\mathcal{L}_\mathrm{j}} \leftarrow \sum_{\mathbf{i=1}}^{\mathbf{m}} \boldsymbol{\beta}_\mathbf{i} \mathbf{f}_\mathbf{i}.(\mathbf{v}(\mathbf{C}))$

            End if

        End for

        Listfound $\leftarrow$ True

        Break {For loop}

    End if

    End for

    If Listfound is false then

    $\mathcal{L}_{|\mathrm{E}^*|+1} \leftarrow \mathcal{L}_{|\mathrm{E}^*|+1} \cup (\mathcal{P}^*, \mathcal{N}^*)$

End for

ListsSearch$(1, \mathcal{L}_\mathrm{j}, \emptyset)$

---------------------------------------------------------------

**Algorithm 5-3: ListsSearch**$(\boldsymbol{\theta}, \boldsymbol{\mathcal{L}}, \mathbf{CS})$

For $(\mathcal{P}^*, \mathcal{N}^*) \in \mathcal{L}$ do

    For $\mathrm{C} \in \mathbb{C}(\mathcal{P}^*, \mathcal{N}^*)$ do

        $\mathrm{CS}' \leftarrow \mathrm{CS} \cup \{\mathrm{C}\}$

        If $(\bigcup \mathrm{CS}' = \mathrm{E})$ and $(|\mathbf{CS}'| \in \boldsymbol{k})$ then

            If $\boldsymbol{V}(\mathbf{CS}') > \mathbf{V}(\mathbf{CS}^*{}_{\boldsymbol{t}})$ **then** $\mathrm{CS}^*{}_t \leftarrow \mathrm{CS}'$ {update $\mathrm{CS}_t{}^*$}

End if

Else

If $(\boldsymbol{\theta} \leq |E^*|)$ and $(V(CS') + \sum_{\mathbf{m=\theta+1}}^{\mathbf{|E^*|+1}} \mathbf{UB}^{W}{}_{\mathcal{L}_m} > \mathbf{V}(\mathbf{CS}_t{}^*))$

{nodes in $CS'$ to -Ve constraints}

$\mathcal{L}' \leftarrow \mathcal{L}_{k+1}$

For $(\mathcal{P}^*, \mathcal{N}^*) \in \mathcal{L}'$ do

For $C' \in CS'$ do

For $e_i \in C'$ do

$\mathcal{N}^* \leftarrow \mathcal{N}^* \cup \{\{e_i\}\}$

End for

End for

End for

$\text{ListsSearch}(\boldsymbol{\theta} + 1, \mathcal{L}', CS')$

End if

End if

End for

End for

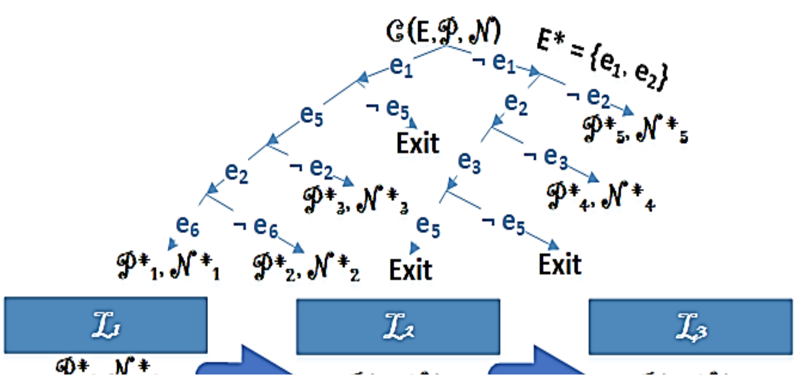



Figure 5-6: DFog-DRF scheme - Joint policy search order for efficient gateway and low-level controller's placement (searching through sorted DRF cluster lists).

Specifically, DRF cluster lists constructed in this way means every feasible joint gateway and low-level controller's placement strategy profile has exactly one link cluster from $\mathcal{L}_1$ and at most one DRF cluster from $\mathcal{L}_i$ , $i > 1$. As such, the algorithm selects a wireless DRF cluster say $C_1$, from some base case in $\mathcal{L}_1$ and checks whether $\{C_1\}$ is a feasible strategy profile or policy i.e., if $C_1 = E$. If this is not the case, then router agents or nodes in $C_1$ are added to the negative co-forwarding constraints of all base cases in $\mathcal{N}^*$ for each $(\mathcal{P}^*, \mathcal{N}^*) \in \mathcal{L}_2$. This action further introduces constraints on the co-located links in that base case, i.e., $\mathcal{L}_2$, ensuring that there is no intersection with wireless DRF cluster $C_1$. In the next step, the fault tolerant control method chooses for every link cluster say $C_2$ in the (now modified) base case in list $\mathcal{L}_2$ and determines if $C_1 \cup C_2 = E$ (a feasible placement policy or strategy profile) If this is not the case, then the process is repeated for $\mathcal{L}_2$ and so on. This means the nodes in $C_1 and\ C_2$ are added to $\mathcal{N}^*$ for each $(\mathcal{P}^*, \mathcal{N}^*) \in \mathcal{L}_3$ . Following this approach, all feasible fog gateway and low-level controller's placement strategy profiles would be examined eventually. The Dfog-DRF scheme implements a branch-and-bound technique (see line 8 in Algorithm 5-2) to speed up the joint policy control process.

## 5.5 Summary

In hierarchical fog networks, traffic emanating from wireless transmission devices for IoT is aggregated and forwarded through gateways. Placing and connecting IoT gateways to the wireless 'backbone' fog devices is critical to efficient operation and management. This chapter proposes a novel decentralised control methodology for gateway and low-level controller's placement optimisation to guarantee improved throughput for IoT computation offloading in wireless hierarchical fog networks.

- ✓ Decentralised fog link scheduling for joint forwarding in hierarchical (wireless) fog-mesh networks that can efficiently support (via constraint dimensionality reduction) bottleneck fairness with dynamics.
- ✓ A model for ground truth interference-aware feature selection of gateways (master nodes) that applies node forwarding strategy (adaptive) feedback to generate fairness

(positive and negative co-forwarding) constraints for the real-time scheduling problem.

- ✓ Application of Map-Reduce framework using unsupervised learning (divide and conquer) to support fast search of fog gateway and low-level controller's placement policies (or strategy profiles).
- ✓ A recursive multi-agent reinforcement learning (MARL) with robustness against adversarial controllers to implement Pareto-improvements using lexicographic multi-objective optimisation to determine cross-layer throughput maximisation of fog gateway and low-level controller's placement policy.
- ✓ The challenging fog link (throughput) scheduling with ***bottleneck fairness*** problem for joint multitask and multi-resource forwarding is unaddressed in literature. This study provided a formal hierarchical co-optimisation model, technological SDN framework, and scheduling scheme to shape hierarchical fog network behaviour (i.e., throughput-awareness and complexity) taking into consideration hierarchical and distributed node forwarding strategy feedback.

# Chapter 6 Performance & Evaluation

- Testbed's design and simulation of experiments to evaluate the effect of strategic feedback control for rejuvenation of decentralised SDN controller deployments in the federated edge using CloudSim and Wireless fog-mesh network using iFogSim.
- Complexity/Scalability analysis of the centralised policy control mechanism for IP-DRF and Dfog-DRF schemes in joint multi-task offloading/forwarding with other state-of-the-art methodology – Asset fair and CEEI-based policies (i.e., Distributed DRF).
- **Simulation results I:** Empirical evaluation of the hybrid scheduling mechanism for IP-DRF (**in chapter 4**) for joint offloading at the federated edge considering fault-prone settings or population and resource non-monotonicity – We compare the allocation policy of D-DRF scheme (Pareto-Optimal) with Second-Best Pareto optimal (SBPO) ones.
- **Simulation results II:** Empirical evaluation of the DFog-DRF scheme for bottleneck fairness or throughput maximisation of fog gateway and low-level controller's placement (joint) policy in the wireless hierarchical fog. Experiments are carried out using iFogSim to show its efficacy based on network average latency and scalability (network use) as important objectives for throughput maximisation.

## 6.1 Experiment's design

### 6.1.1 Federated edge testbed

**CloudSim Features.** The key features of CloudSim are described in Figure 6-1 and Figure A-0-1 [184] which allows for modelling system and complex behaviour of components in Cloud computing. Exploring dynamic resource allocation mechanisms in edge and Fog environments that are massively distributed and scalable, the simulation of clouds and applications is applied to evaluate performance and further provide useful insights.

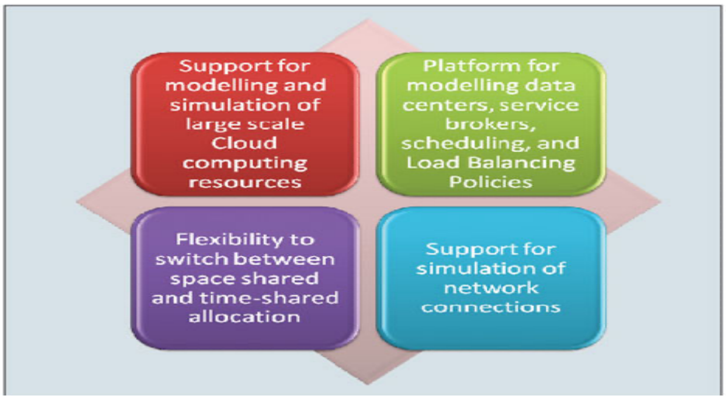



Figure 6-1 CloudSim Features

Implementing the core functionality required for higher-level simulation frameworks is the SimJava discrete event simulation engine [184]. The engine is at the lowest layer, such as communication between components (see Figure A-0-2 [184]), creation of system components (services, host, data centre, broker, and VMs), and management of the simulation clock, queuing, and processing of events.

**FEC Data centres.** The simulator models core hardware infrastructure for cloud-related services using the Data centre Class. Essentially, infrastructure level services i.e., hardware and software, provided by resource providers in a Cloud environment are modelled using this class. The Data centre class comprises several compute hosts that are homogeneous or heterogeneous as regards to capacity, cores, memory, and storage. In addition, every Data centre creates an instance of a generalised resource-provisioning component that supports policies for allocating memory, bandwidth, and storage devices.

**Decentralised SDN controller (Edge ResourceLayer** [43]**).** This class models a DRF broker, whose responsibility for fair allocation between FEC service providers and FEC users. However, the broker functions to establish user QoS requirements and deploy service tasks amongst Clouds as shown in Figure 6-2. The Edge broker acts on behalf of users to identify via the Cloud Information Service (CIS) appropriate Cloud service providers. The broker than proceeds to negotiate with them for an allocation of resources that meets QoS needs of users. We extended this class to conduct experiments with A custom developed allocation policy (non-intersecting or non-overlapping).

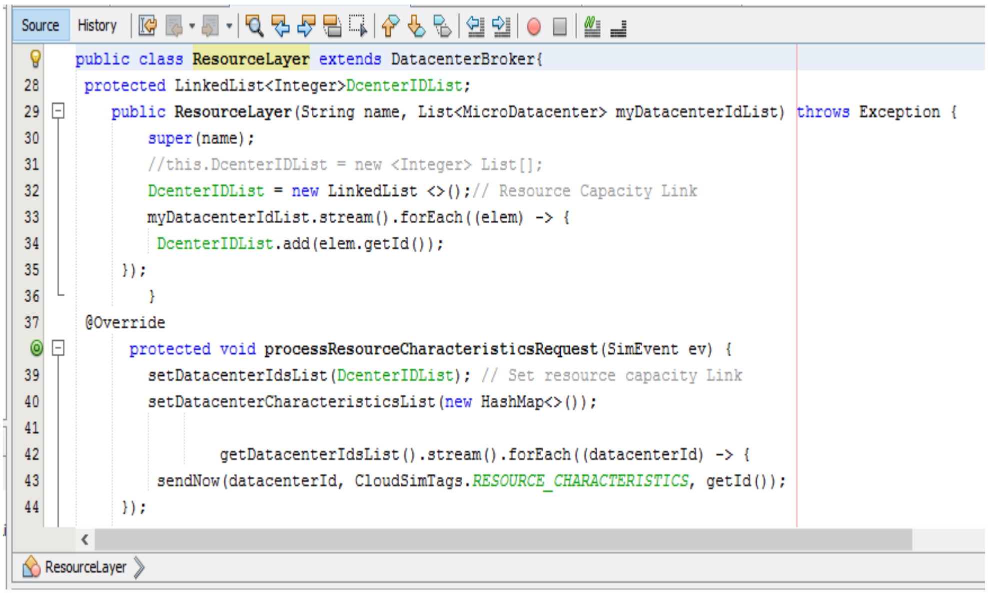



Figure 6-2: Edge Datacenter Broker – decentralised SDN controller implementation

### 6.1.2 IFogSim Testbed

The simulation kernel for IFogSim [34] is an extension of CloudSim, therefore, the experiments design would focus on the implementation of testbed components.

We provide a description of the elements of network architecture using the classes of iFogSim is shown in tables 6-8 and 6-9. The implementation of the test bed for Wireless fog mesh network environment using iFogSim consists of simulated entities and services.

#### 6.1.2.1 Physical Elements

**Edge/Fog Device.** In this class, all characteristics of hardware for fog devices (i.e., mobiles, WIFI gateways, proxy, and cloud) and their connections to other actuators, fog devices, and sensors is given. The definition of the communication capacity of fog devices i.e., downlink bandwidth, memory, processor, storage size, uplink, are the key attributes of the FogDevice class. This class allows researchers to define how the fog device resources are scheduled between application modules running on it and how modules are deployed and removed. Overriding these methods allows for plug-in custom policies. Essentially, the realisation of this class is an extension of CloudSim's PowerDatacenter class.

**IoT Sensor.** An instance of the sensor class represents an IoT sensor in the network architecture. In this class, the characteristics of a sensor, from connectivity to attributes for output is contained.

The class contains a reference attribute to the gateway fog device to which the sensor is connected and the latency of connection between them. Most importantly, the output characteristics of the sensor and the distribution of tuple inter transmission is defined in this class, which identifies the tuple arrival rate at the gateway. By setting appropriate values of these attributes, devices like smart cameras and connected cars can be simulated.

**IoT Actuator.** The model of an actuator is defined by the effect of actuation and its network connection properties. The class defines an override method to perform, at basic levels, an action on arrival of a tuple from an application module and can be extended to implement custom effects of actuation. An attribute in the class refers to the wireless gateway (Fog Device) to which the actuator is connected and the latency of this connection.

**Data Stream Layer (Tuple Class).** A realisation of the data stream layer in the wireless hierarchical network architecture and the basic unit of communication between entities in the Fog. Inherited from the Cloudlet class of CloudSim is the Tuple class. The characteristics of a tuple is its type and the source and destination application modules. Attributes of the class specify the length of data encapsulated in the tuple and the processing requirements defined as million instructions (MI).

#### 6.1.2.2 Logical Parts

**Smart City Application.** The application design in the experiments follows the Distributed Data Format (DDF) model, whereby applications are modeled as a directed graph. Modules represent the vertices of the directed acyclic graph (DAG) that perform processing on incoming data and edges denote the relationships (data dependencies) between modules. These entities are realised using the following classes.

- **AppModule.** This class is realised by extending the class PowerVm in CloudSim. For each incoming tuple, an AppModule instance processes it and generates output tuples that are sent to next modules in the DAG. Using a selectivity model, the number of output tuples per input tuple is chosen—which can be bursts or fractional. Instances of AppModule class represent processing elements of fog applications and realise the vertices of the DAG in DDF model.
- **AppEdge.** An instance of this class denotes the data dependency between a pair of smart city application modules and represents a directed edge in the DDF application model. Captured by the tupleType attribute of AppEdge class, each edge is typified by the type of tuple it carries. The processing requirements and length of data encapsulated are also captured in this tuple. iFogSim supports periodic and event-based application edge; For the tuples on a periodic AppEdge, these are emitted at regularised intervals. Furthermore, tuples on an event-based edge e = (u, v) are sent when the source module u receives a tuple, and the selectivity model of u allows the emission of tuples carried by e.

#### 6.1.2.3 Management and Monitoring

**Control Loop.** AppLoop Class is used for specifying the process-control loops of interest to the user. For the wireless fog mesh networks, the control loops are specified to measure the end-to-end latency. A list of modules beginning from the origin of the loop to the module where the loop ends constitute an AppLoop instance. In addition to the basic tuple processing functionality, simulated services available in iFogSim are as follows:

- **Resource Management Services.** The Fog simulator implements placement and scheduling of resource management for IoT applications. These abstract classes are independent of each other to facilitate easy customisation and extension.
- **Application Placement.** After submission of applications for processing, the placement policy determines how application modules are placed across Fog devices.

**Monitoring service.** Each device monitors and maintains its current resource use statistics in the current version of iFogSim. The *executeTuple () method* in the FogDevice class contains the tuple processing logic whereby the device updates its resource usage. The statistics can also be stored in a tuple and pushed to the resource management layer for running use-aware resource management policies. This information can be useful in the studying the performance of the IoT application on fog infrastructure and may be collected as logs to be analysed offline.

## 6.2 Simulation and discussion: Study I

In this sub-section, we perform several simulated experiments and present empirical results to (1) compare the deterministic policy of the hybrid scheduling mechanism provided by IP-DRF to allocation policy of the D-DRF scheme (see **Table 6-1**) (2) compare the running time and accuracy of the IP-DRF scheme with other state of the art methodology for solving similar problems - the results from simulations [133] , [158] for the IP control algorithm highly correspond to results we obtained from similar experimental setup.

TABLE 6-1: TAXONOMY OF FAIR ALLOCATION POLICIES FOR RECONFIGURING THE FLAT SDN CONTROL PLANE

| Allocation Property | Asset Fairness (Centralised) | CEEI (D-DRF) (Decentralised) | IP-DRF (Decentralised) |
|---|---|---|---|
| Sharing Incentive | | Y | Y |
| Group strategy Proof | Y | | Y |
| Population & Resource Monotonicity | Pop. | | Y |
| PO/SBPO | PO | PO | SBPO |

### 6.2.1 Complexity analysis of centralised policy control mechanism

**Run Time & solution quality.** For comparative purpose only, two state of the art algorithms for fair allocation namely integer programming (based on ILOG's CPLEX) and Improved dynamic programming $O(3^n)$ for CSG are considered. Using sub-additive, super-additive, normal and uniform distributions used to generate test cases for the Nash bargaining solution by [165], [185], the policy control mechanism of IP-DRF scheme always finds the best allocation policy quickest compared to the rest of the solutions. In the worst case, the IP-DRF (anytime) scheme computes the solution for ***27 edge nodes*** in 1.3 hours or $4.69 \times 10^3 seconds$ while the DP algorithm waits approximately $5.2 \times 10^6\ seconds$. In the best case, the centralised policy (IP) control mechanism of IP-DRF scheme only needs $2.2 \times 10^{-5}$ % of the time for DP algorithm in the CSG – these metrics are much smaller when using graphic processor units (GPUs). For population non-monotonicity setting, with changing number of singleton clusters, the growth rate of the run-time for IP-DRF's deterministic policy considering $n$ controller nodes can be proven to be approximately $O(2.7^n)$ given an NCDS distribution and like $O(2.3^n)$ given uniform and normal distributions. This is significantly less than $O(n^n)$ growth rate of the size of the flat distributed SDN Policy space.

**The ratio bound $\boldsymbol{\beta}$ (as a function of searched Joint DRF policies)**. In CSG literature, [165] and [185] are the two algorithms that improve solution quality. The proposed IP-DRF scheme is benchmarked against these two schemes for evaluation. With 24 singleton nodes in the free rider problem or resource non-monotonicity, the number of strategy profiles required IP-DRF to establish a bound $\boldsymbol{\beta}$ of 3 is 0.0012% that of [165] and approximately 1% that of [185]. On the other hand, given a congestion problem setting or population non-monotonicity, the IP-DRF's hybrid scheduling mechanism for joint offloading requires searching more strategy profiles (Joint DRF policies) as compared to other scheduling algorithms or mechanisms to establish its first ratio bound (definition 4.4.1). However, once the search is done, the number of strategy profiles required to establish a bound of 3 is only approximately 0.00067% as compared to the [165] and 0.51% of [185].

Table 6-2: Consumer task instances and resource requirements at the platform edge.

| Bid Type | Length | **File Size (MB)** | **Output Size** | **PE** | **Util. Mode** | VM Priority | MIPS | **Image Size (MB)** | **RAM (MB)** | **Bandwidth** | **CPUs** | **VMM** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\theta_{\mathbb{B}}^{LVM}$ | 5000 | 30000 | 30000 | 2 | Full | I | 8000000 | 40000 | 4096 | 10000 | 2 | Xen |
| $\theta_{\mathbb{B}}^{SVM}$ | 7000 | 50000 | 60000 | 1 | Full | II | 800000 | 4000 | 2048 | 10000 | 1 | Xen |

### 6.2.2 Simulation parameters

***The joint offloading model at federated edge***. Offloading bids are allocated a share of processing power on Data centre's host components; application elements are sandboxed within VMs or container engines (CE). Host is a component that represents a physical computing node in an EC: a pre-configured processing (expressed in millions of instructions per second– MIPS, per CPU core), memory, storage, and a scheduling policy for allocating processing cores to virtual machines. Administrative EC set $\mathbb{C}$ assigned a different compute capacity (of MIPs) and $cap(\rho_i) \in \mathbb{R}^+$ in the marketplace using settings of $\rho_i$= ki×108. With $k = \{2.0, 3.0, 4.0, 5.0\}$ representing the order (multitude) of available compute capacity $k_i \in \mathbb{R}^+$ in data centre host lists $H_i$ (with 4 PEs per host) in the federated edge.

The Cloudlet class is extended to create the IoTCloudlet class (used to create offload task list $\Theta_{\mathbb{B}}$) whose instance represents the offload application service activation $\theta$. ***Table 6-2*** lists the simulation parameters for two consumer offloads $\{\theta^L(I), \theta^S(II)\}$ including the EdgeVm/container priority, task length, file size, output size, and processing elements (PEs). Logically distributed SDN controllers or DRF-enabled clusters [6], [85] are created from the EdgeResourceLayer [43] class which extends the DatacenterBroker class in the CloudSim toolkit [20]. Each Host component instantiates a VM/Container scheduler component that implements time-shared policies for allocating cores to VMs/containers. The VM class is extended to include a VM/Container component characteristic related to its weight or scheduling priority (i.e., I or II).

#### 6.2.2.1 Population non-monotonicity

**User Offloading Bids.** Offloading agents $B_1, B_2, B_3$ and $B_4$ each receive IoTCloudlet instance bids generated from the uniform distribution in Table 6-3.

TABLE 6-3: SETTINGS FOR THE SIMULATION OF POPULATION NON-MONOTONICITY

| Byzantine Strategies | Bids {I, II}<br>Uniform distribution |
|---|---|
| Lose-lose | $\theta_{B_2}$= {[4,500, 4,600], [2,400, 2,500]} |
| $[C_4, C_2]$ | $\theta_{B_4}$= {[4,500, 4,600], [7,000, 7,145]} |
| Win-Win<br>$[C_1]$ & $[C_3]$ | $\theta_{B_1}, \theta_{B_3}$= {[16,500, 17,000], $\emptyset$} & $\|e_{t,\bar{\varsigma}}\| \le 0$ |

Win-lose (Distributive) strategies – With the advantage of geographic proximity, data centre agents $C_4$ adopts a win-lose VM migration (zero-sum) strategy with EC agent $C_2$ to correct provisioning mistakes from production function $\mathcal{F}_4(\theta^{LVM}, \theta^{SVM})$ because of temporary host outages $h_{4z} \in H_4$.

Win-Win (Integrative) strategies – ICGs with negative settings. Administrative data centre agents $\mathrm{C}_1$ and $\mathrm{C}_3$ face data centre host failures (e.g., Fires or floods) and adopt integrative strategy to handle excess offloads $\theta_{\mathbb{B}}{}^{LVM}$. In socio-economic interaction for experiments, ICGs (interactive strategies) can only lead to negative zero-sum bargain externalities i.e. $\|\mathrm{e}_{\mathrm{t},\bar{\varsigma}}\| \leq 0$ or decrease the availability of external resources.

### 6.2.2.2 Resource Non-Monotonicity

**Bids ($\mathcal{F}$) and Non-excludable Bids**. Table 6-4 describes experiment setup for IoTCloudlet instances requested at offloading agents $B_1$, $B_2$, $B_3$ (includes paid bids and non-excludable bids), $B_4$.

Table 6-4: Settings for the simulation of Resource Non-Monotonicity

| Byzantine Strategies | Paid-Bids - $\mathcal{F}$ {I, II} Uniform distribution | Non-Excludable Bids {I, II} Uniform distribution |
|---|---|---|
| Win-lose $[\mathrm{C}_3, \mathrm{C}_4]$ | $\theta_{B_4}$ = {[3500, 3,600], [4,500 4,700]} | $\theta_{B_3}$= {∅,[8,400, 8,575]}, > $\mathcal{F}_3(\emptyset, \theta_{\mathbb{B}}^{SVM})$ |
| Win-Win $[\mathrm{C}_1]$, $[\mathrm{C}_2]$ & $[\mathrm{C}_4]$ | $\theta_{B_1}$= {[3500, 3,600], [2,500, 2,860]}, $\theta_{B_2}$ = {[3500, 3,600], [3,500, 3,810]} $\theta_{B_3}$= {[7,000, 7,200], ∅} | $\|\mathrm{e}_{\mathrm{t},\bar{\varsigma}}\| \geq 0$ |

**Distributive EC strategies.** Because of the uncertainty in production $\mathcal{F}_3(\theta^{LVM}, \theta^{SVM})$ from the behaviour (bidding) of consumer agent $B_3$, cloud or data centre agents $C_3$ and $C_4$ adopt (positive) zero-sum bargain strategies to support capacity scheduling and allocation of unpredictable offloads.

**Integrative EC strategies for DRF broker (main) and secondary controllers' –** ICGs with zero-sum settings. Essentially, EC contractual or win-win strategies are solely meant to support Platform innovation – In the socio-economic interaction during experiments, ICGs (integrative strategies) can only cause positive controller-to-controller bargain externalities $\left\|\mathrm{e}_{\mathrm{t},\bar{\varsigma}}\right\| \geq 0$ or increase the availability of non-excludable resources $\theta_{\mathbb{B}}^{SVM}$.

***FEC Marketplace - Cost Model***

Modelling of costs and pricing policies is an important aspect to be considered in the simulation of Cloud systems. Two basic pricing policy for offloading processing cost are used to evaluate the IP-based platform governance, these are:

- Fixed cost. The cost per unit processing in all data centres is fixed.
- Production-based (variable) cost. In this market scheme, set $w_i$ the unit processing cost in EC data centre i offering capacity $\rho_i$ in MIPS. The MIPs available is directly proportional to the cost per unit processing $w_i \propto \rho_i$. For simplicity, this proportionality is linear, as such let $w_i = \sigma.\rho_i$.

Where $\sigma = {}^{w_i}/_{\rho_i}$. In all experimental runs, we set $\sigma = 1 \times 10^{-7}$ for each of the 4 fog infrastructures (data centres) using the production-based cost scheme, and $w_i = 20$ for the fixed cost. All experiments were carried out on a 32-bit OS (x64 based processor) machine with specification Intel Duo core T660 @ 2.20 GHz and 3.0 GB installed RAM.

## 6.2.3 Fault-tolerance, monotonicity, and rejuvenation of the flat SDN plane

The knowledge of Controller-to-controller bargain externalities is calculated (for each DRF coalition) as ${\mathrm{e}_{\mathrm{t},\bar{\varsigma}}}^{\mu} = \mu\left\|\mathrm{e}_{\mathrm{t},\bar{\varsigma}}\right\|$ where $\mu \in (0,1]$, Second-best Pareto-optimality (SBPO) assumes perfect information or 100% knowledge of externalities. Using exact test method in analysis of the truthful mechanism of IP-DRF, we set the weights of preference criterions for ***fault-***

***tolerance*** as $\alpha_1 = 0.99, \alpha_2 = 0.01$ and compare first best PO structures of Nash bargaining based allocation policy called distributed DRF [132] with SBPO ones. The IP-DRF scheme provides a deterministic policy that guarantees second-best Pareto optimality (SBPO) in resource allocation for IoT cloudlets. Figures 6-3 and 6-4 depict the efficacy of IP-DRF scheme with increasing knowledge of bargain externalities ($\mu = 10\%$ to 100%) in both positive and negative zero-sum Controller-to-controller bargain settings for ICG. With much higher (e.g., 80-100%) knowledge of zero-sum or controller-to-controller bargain values (externalities), the hybrid scheduling policy mechanism of IP-DRF scheme exhibits strong rejuvenation and reconfiguration capabilities or fault proof characteristics.

**Objective 1: Execution time and Turnaround time ($\boldsymbol{f_1}$). Table 6-5 (and Figure 6-5 & Figure 6-7)** shows at 100% knowledge of Controller-to-controller bargain externalities (Second-best) how **IP-DRF**'s policy outperforms that of the D-DRF scheduling scheme. The average time for processing of 57,445 bids (IoT cloudlets) for both execution and response time is reduced by 6.89% and 23.96% in resource non-monotonicity settings and 19% and 11.52% (respectively) in the population non-monotonicity problem. In particular, the second-best PO policy achieved by IP-DRF scheme reduced the relative average execution time for first best PO (the D-DRF scheme) by 0.0624secs and average turnaround time by 0.0542 secs for the population non-monotonicity setting. For resource monotonicity, the IP-DRF scheduling scheme outperformed D-DRF scheme by reducing the average execution and turnaround time by 0.0276s and 0.1682secs.

**Objective 2: Processing Cost ($\boldsymbol{f_2}$).** ***Figure 6-6 & Figure 6-8*** shows for the SBPO policy, output of the IP-DRF offloading scheme (in Chapter 4) also performs better than the D-DRF scheme in minimising cost for both the fixed and variable pricing strategy in ICGs with environment settings of population and resource non-monotonicity for multitask offloading. The average cost of processing all 38,000 IoT cloudlets (bids) using both fixed cost and variable cost pricing strategy is reduced by 6.89% and 9.03% in resource non-monotonicity settings and 19.1% and 12.73% in the population non-monotonicity settings respectively.

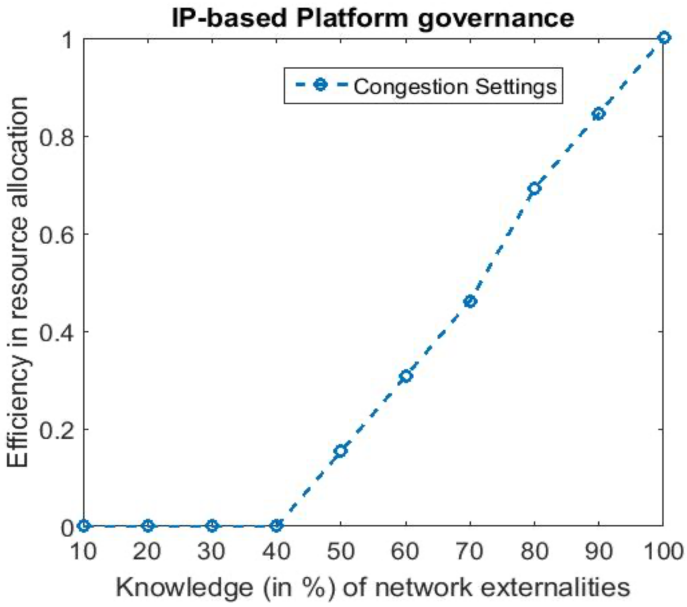



Figure 6-3 Robustness against population non-monotonicity setting

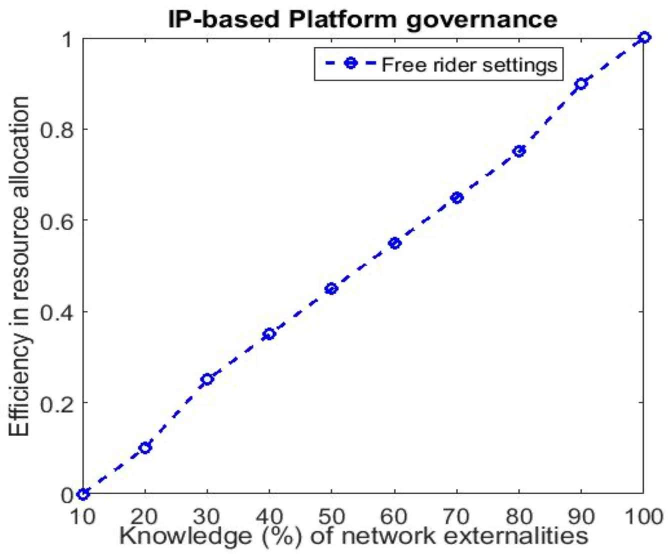



Figure 6-4 Robustness against resource non-monotonicity setting

Table 6-5: Comparing Pareto-optimal (PO) structures with Second-best ones SBPO.

| **Experiment settings (Network Effects)** | **Strategic Game Solution** | **SLA Deadline Violation (0.51s/0.6s)** | **Reduced Ave. Exec. Time (Relative)** | **Reduced Ave. TAT (Relative)** | **Reduced Processing Cost (Relative in %)** | |
|---|---|---|---|---|---|---|
| | | | | | Fixed | Variable |
| **Congestion (57,445 cloudlets) Resp. time=0.51s** | First-Best PO ICG | 1,488 jobs (2.6%) | - | - | - | - |
| | Second-Best PO ICG (-Ve) | 0 (0%) | 0.0624s (19%) | 0.0542s (11.52%) | 19.1% | 12.73% |
| **Free Riding (38,000 cloudlets) Resp. time=0.6s** | First-Best PO ICG | 38,000 jobs (100%) | - | - | - | - |
| | Second-Best PO ICG (+Ve) | 12381 jobs (32.58%) | 0.0276s (6.89%) | 0.1682s (23.96%) | 6.89% | 9.03% |

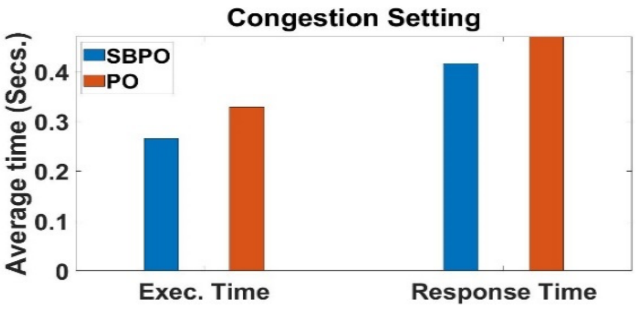



Figure 6-5 comparing SBPO with PO deployment for population non-monotonicity setting (time)

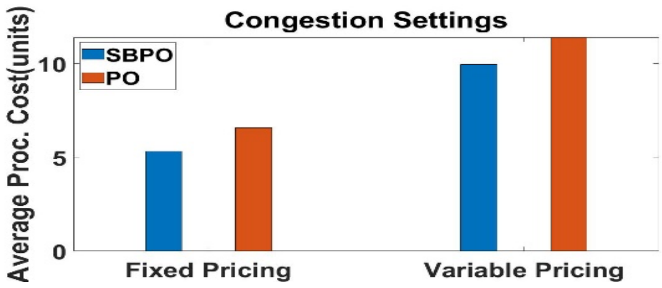



Figure 6-6 comparing SBPO with PO deployment for population non-monotonicity Setting (cost)

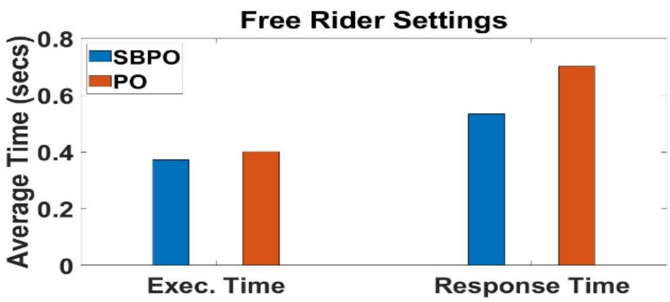



Figure 6-7 comparing SBPO with PO deployment for resource non-monotonicity setting (time)

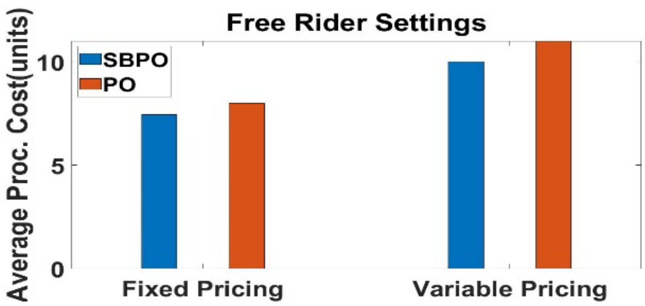



Figure 6-8 Comparing SBPO with PO deployment for resource non-monotonicity setting (cost)

## 6.3 Simulations and Discussion: Study II

In this sub-section, it is necessary for dynamic hierarchical fogs to empirically evaluate the DFog-DRF scheme against other centralised and decentralised methods for proactive rejuvenation of controller deployment (See **Table 6-6**) taking into consideration: (1) benchmarking or run-time optimisation against the state-of-the-art methodology for fair allocation in literature – the results correspond to similar experiments [134] performed (2) the throughput awareness of Dfog-DRF forwarding scheme as compared to Nash bargaining (distributed) DRF scheme for the EEG tractor beam game of iFogsim.

Table 6-6: Taxonomy of Fog Link scheduling schemes for reconfiguration (with bottleneck fairness)

| Allocation Property | Asset Fairness **Centralised** | **D-DRF decentralised** | **DFog-DRF decentralised** |
|---|---|---|---|
| **Load balancing** | N | Y | Y |
| **Group strategy proof** | Y | N | Y |
| ***Pareto-Optimality*** | *Y* | Y | Y |
| ***Bottleneck fairness*** | *N* | Y | Y |

### 6.3.1 Run-time complexity analysis of centralised policy control mechanism

The hybrid scheme for gateway and low-level controller's placement optimisation were benchmarked against a modified version of the mixed integer programming formulation for the dynamic equilibrium problem, which has a column for every feasible wireless DRF cluster, instead of a column for every possible wireless DRF cluster. Empirical evidence shows that DFog-DRF scheme outperforms by *orders of magnitude* the mixed integer programming (MIP) scheme [186] for optimisation of the hierarchical WMN. The DFog-DRF scheme solves the optimisation problem instances comprising 20 and 30 physical SDN nodes and for 100 and

1000 positive and negative fairness constraints. Given a different number of edge resource nodes, we perform a comparison between the times the generalised algorithm takes to generate the feasible clusters (using the transformation process) and the search for feasible wireless DRF clusters by going through every coalition in $2^E$ against the specified constraint to check whether it is feasible or not. The proposed Dfog-DRF scheme is faster by orders of magnitude - for instance, given 30 router nodes, the fault tolerant control mechanism takes a fraction of the time (< 1%) for both r=100 and r=1000 wireless DRF cluster constraints.

***Fog gateway and low-level controller's placement (joint) policy***. For evaluation purposes, we compare the hybrid mechanism of Dfog-DRF scheme which applies node forwarding strategy feedback with two other mechanism designs [187], [186] given the same input parameters. **Nash Bargaining solution** is implemented such that it checks every possible cluster against the list of QoS constraints storing only feasible ones. Here, the centralised policy control mechanism proceeds as normal except that every branch in the search tree is pruned if it does contain a cluster that is not feasible. The mixed integer programming formulation for synergy coalition groups was implemented using a standard MIP package (ILOG's CPLEX). Given a normal distribution and different number of transmission or forwarding controllers, the proposed control algorithm outperforms all other (state-of-the-art) mechanisms by orders of magnitude. Table 6-7 shows the performance of the generalised algorithm considering 30 transmission nodes. The average number of DRF clusters was nearly $8 \times 10^7$ and $6 \times 10^8$ given 100 and 1000 pairwise-fairness constraints respectively.

Table 6-7: Relative comparison (speed) of Dfog-DRF scheme with other scheduling schemes (20 & 30 nodes)

| Algorithm | Edge resource nodes | r=100 | r=1000 |
|---|---|---|---|
| **Nash Bargaining solution (coalition games)** | **20** | **0.079%** | **0.104%** |
| | **30** | **0.0017%** | **0.38%** |
| **Asset Fairness (MIP-CPLEX)** | **20** | **0.025%** | **0.003%** |
| | **30** | **0.065%** | **0.01%** |

## 6.3.2 Empirical analysis using IFogSim (Latency sensitive online game)

**IFogSim.** An effective tool for evaluating resource management approaches including cloud-only application module placement and edge module placement - a technique that pushes applications towards edge devices when there are enough resources.

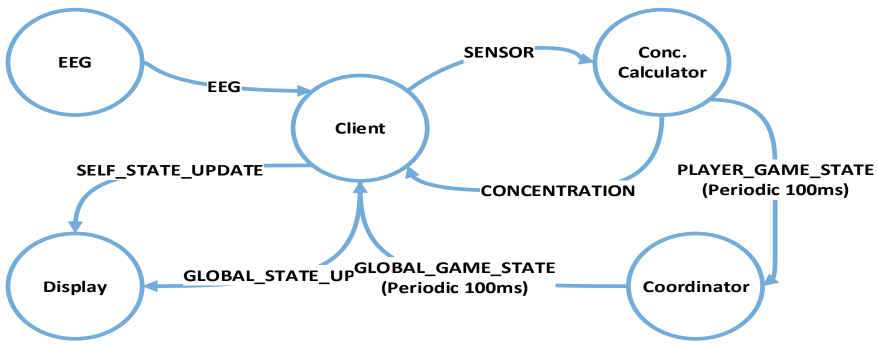



Figure 6-9: Two apps architecture [34]

The latency-critical application model (Figure 6-9) in the case study [34] is a human-vs-human game (EEG Tractor Beam Game) that involves augmented brain computer interaction. To play the game, every player/actor needs to wear an MINDO-4S wireless EEG headset connected to his smartphone. The Tractor Beam Game is implemented as an Android application on the smartphone of users. The application is responsible for real-time processing of the EEG signals sensed by the EEG headset as well as computing the brain state of the user.

The availability of iFogSim should help rapid development of innovative resource management policies in the areas of fog computing and IoT and with end-to-end modelling and simulation.

Table 6-8: Configuration of fog devices for EEG applications

| **Edge Devices** | **CPU (GHz)** | **RAM (GB)** | **Mips ($\times 10^3$)** | **Bandwidth (Up, Down) ($\times 10^3$)** |
|---|---|---|---|---|
| **Cloud (1)** | **3.0** | **8** | **32.8** | **[10, 10]** |
| **WIFI Gateways (2)** | **3.0** | **4** | **4** | **[1 10], [1 10]** |
| **Mobiles (210)** | **1.6** | **1** | **1** | **[0.1, 0.27]** |
| **Proxy-server (1)** | **3.0** | **4** | **16** | **[10, 10]** |

Table 6-9: Simulation parameters of Edge Devices

| Source | Destination | Latency (due to hops) in ms |
|---|---|---|
| IoT Sensor | IoT Device | 6 |
| IoT Device (Mobiles) | WIFI Gateway | [0,2], [2,3], [3,5], [8,10] (uniform distributions) |
| WIFI Gateways | Proxy-server | 4 |
| Proxy-server | Cloud DC | 100 |

**TwoApps.** The critical control loop in the EEG tractor game application (latency of response) is the loop with the responsibility of transforming the brain-state of the user into a game state on the device actuator. This requires real-time communication between the IoT device and the device hosting the brain-state classification module together with efficient processing on the classification module. Delay in this loop will severely affect (negatively) user experience as it affects the entities users interact with.

### 6.3.3 Experiment setup

For rejuvenation, there are several metrics (optimisation objectives) that need to be optimised in fog network system. Performance of the application is related to the amount of time (latency) needed to accomplish a certain task. Also, importance is attached to the economic use of scarce resources (resource usage) particularly in the lower layers of the fog model as this would also minimise power consumption on mobile devices.

**QoS Constraints for fog gateway and low-level controller's placement.** For all seven fog-based edge resource nodes in E from example 5.1, Mesh clients (**MC**) or the number of mobile devices $d_i$ (individual links) connected to the edge resource $e_i$ was set as $|d_i e_i| = \{20, 50, 80, 10, 30, 10, 10\}$ in experiments. The DRF cluster load constraint $\boldsymbol{L_{max}} = 110$**.** The relay load in DRF clusters is simply computed as $ld(C) = \sum_{d_i e_i \in C} |d_i e_i|$and the DRF cluster size constraint $\mathbf{S_{max}} = 4$.

### 6.3.4 Bottleneck fairness and group strategy proofness

**Positive and Negative (pairwise-fairness) constraints**. The 2-stage feature selection and transformation for bottleneck fairness constraints in our experiments are from example 5.1 (in Chapter 5). The DRF cluster delay constraint $R_{max}$ for fog-based Steiner resource trees is set to not more than 2ms (Table 6-8). All pairwise-fairness constraints for joint forwarding used in experiments conform to this criterion. To evaluate the DFog-DRF scheme, we setup experiments using the simulation parameters of edge devices in Table 6-9. Different intervals of uniform distributions are used to randomise the hop distances (in ms) for connection of the IoT devices under low-level controller nodes to gateways. For empirical evaluation purposes, we choose k= 2 WIFI gateways and evaluate Six (6) feasible gateway and low-level controller's placement (joint) policies as fog run-time environments for empirical analysis of the Dfog-DRF scheme. In experiments, fog gateway and low-level controller's placement (joint) policy $GP_1$ of the Dfog-DRF scheme is group strategy proof [6] or provides a truthful mechanism using *pairwise-fairness constraints* from *multi-controller forwarding strategy feedback related to homeostasis*. Nash bargaining solution or distributed DRF policies [132] $GP_2$, $GP_3$, $GP_4$, $GP_5$ and $GP_6$ do not avoid scheduler manipulation – using localised resource managers for coordination. However, the deterministic policy mechanisms for fog gateway and low-level controller's placement guarantee the sharing incentive or load balancing for performance isolation as well as bottleneck fairness for multiple resources. These fog configurations are simulated for two applications per device to implement joint multitasking offloading in the iFogsim testbed. The *hierarchical fog re-configuration $GP_1$* is the Pareto-optimal structure of the centralised scheduling policy control described in **Chapter 5** whereas re-configurations $GP_2$ to $GP_6$ are all feasible ***structures*** of the Distributed DRF scheme implementation in Nash bargaining game settings based on QoS constraints.

### 6.3.5 Cross-Layer throughput maximisation: Rejuvenation

Lexicographical ordering is the method used by Dfog-DRF scheme to maximise throughput in fog gateway and low-level controller's placement (joint) policy providing a fault ***tolerant control plane***. Essentially, latency of the application loops is considered the most important priority ($\alpha_1 = 0.99$) and network usage is considered the least important ( $\alpha_2 = 0.01$).

**Objective 1: Average latency of control loop (using the Edgeward placement strategy).**

Figure 6-10 and Figure 6-11 illustrate the average delay in the execution of the control loops in TwoApps for the 6 feasible strategy profiles. For smaller WIFI gateway bandwidth (1,000MHz), $GP_1$ and $GP_3$ show the most reduced delay for application loop 1 (EEG, CLIENT, CONCENTRATION_CALCULATOR, CLIENT, DISPLAY]) whereas strategy profile $GP_1$ showed significant reduction in average latency for application loop 2 (1.7646secs). With more WIFI gateway bandwidth (10,000MHz), $GP_1$ showed the most reduced average latency

(1.4565secs) for application loop 1 however, strategy profile $GP_5$ reduced the average delay the most for application loop 2 in simulated experiments.

**Objective 2: Network Usage (Edgeward placement strategy).**

It is worth noting that a bulk of communication in this application happens between Client and Concentration Calculator modules. Figure 6-12 depicts the network use of the EEG tractor beam game application for 210 mobiles and 420 applications. The load on the network increases due to an increase in the number of devices connected to the application significantly where only cloud resources used. As shown in Figure 6-12, when different gateway bandwidth (1,000 and 10,000) is considered, the network use ($0.2669 \times 10^6$ and $1.0980 \times 10^6$) considerably decreased using strategy profile $GP_1$ in comparison to all other feasible D-DRF policies. This result also gives a clear perspective regarding the scalability of fog-based applications. Allowing the network use to grow uncontrollably as in case of cloud-based execution can lead to network congestion and further degradation of the performance of applications. Such situations can be better avoided if fog-based deployment is adopted, and information is pre-processed closer to the source of data.

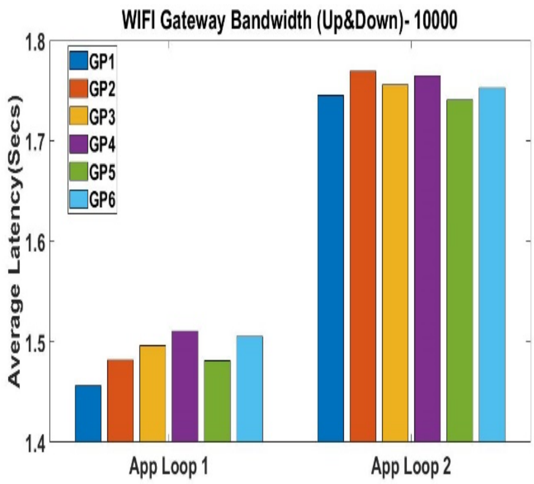



Figure 6-10. Evaluation of response time (inverse throughput) for GP strategy profiles (10,000 BW)

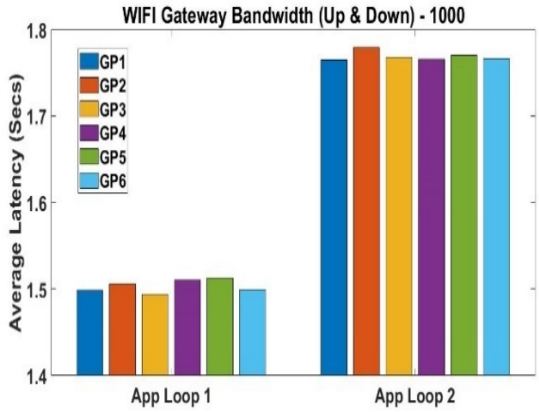



Figure 6-11. Evaluation of response time (inverse throughput) GP strategy profiles (1,000 BW)

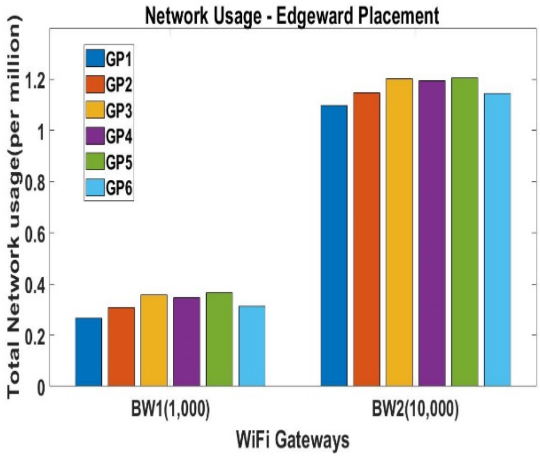



Figure 6-12. Scalability - Evaluation of Network bandwidth usage (cost)

## 6.4 Summary

Experiments using simulated testbeds show: (1) improved performance for our novel hybrid scheduling schemes that applies feedback in scheduling for IoT cloudlet instances as compared to the hybrid scheduling scheme of distributed DRF or D-DRF (2) The proposed MARL policy for joint forwarding is faster than the mixed integer programming implementation (RL-based) and the Nash bargaining based (distributed) DRF scheme to solve the same problem.

- ✓ In this chapter, critical evaluation of the hybrid scheduling policy or truthful mechanisms for joint offloading at the distributed SDN control plane of the federated edge and hierarchical fog (wireless) networks is carried out.
- ✓ The setup, results and discussion of the experiments is provided for joint multitask offloading considering (1) population and resource monotonicity scheduling problems of the federated edge and (2) the fog scheduling with bottleneck fairness problem in hierarchical (and dynamic) wireless networks.

# Chapter 7 Conclusions, Limitation and Future Work

## 7.1 Conclusions

Due to the increase in the request of the cloud users particularly with the IoT, fair resource allocation in next generation cloud computing or Data centre environments (like FEC) has been progressively become one of the critical issues. Addressing these FEC scheduling problems for joint offloading needs an effective mechanism design that could guarantee user satisfaction in terms of service level agreements (SLAs) and QoS in particular. The two research questions, related work, and thesis methodology for the design of SDN-based resource allocation mechanisms in distributed data centre environments for FEC applications are discussed in preliminary chapters (**Chapters 1,2 &3**) of this thesis.

This thesis described in **Chapter 4**, the rejuvenation of decentralised controller deployment from fair scheduling and allocation of joint offloads at the federated edge considering two distinct sides (provider and consumer). The fault tolerant flat SDN control plane scheme (using MARL) devised in **Chapter 4** incorporates centralised Joint DRF policy control mechanism applying distributed node offloading strategy feedback related to homeostasis to form the basis for rejuvenated decentralised controller deployment at the federated edge. Also addressed in the part study in **Chapter 4** was the hybrid monotonic scheduling framework for IoT offloading in population (the congestion problem) and resource (free rider problem) non-monotonicity environments at the federated edge. By proposing an incentive-compatibility mechanism (algorithmic) for governance, harmful negative and positive Controller-to-controller bargain externalities leading to population and resource non-monotonicity are neutralised. To achieve this, work applies specific theorems in CSG literature (definition 4.4.1 - definition 4.4.7) using lexicographic ordering to provide a mechanism that is fault-tolerant and provides Second-best

Pareto optimal (SBPO) allocation which outperform Nash bargaining solution (Pareto optimality). We proposed in this thesis, the IP-DRF scheduling scheme to ensure second-best Pareto optimality in joint offloading for both population (congestion) and resource (free-riding) non-monotonicity settings. Taking into account sub-additive, super-additive, normal and uniform distributions used to benchmark Nash bargaining (CSG) algorithms in [165], [185], the IP-DRF scheduling scheme was designed to find the best scheduling policy (Pareto-optimal) quickest compared to the rest of the schemes. With 24 SDN-based edge nodes in the resource non-monotonicity problem, the number of strategy profiles required by IP-DRF scheduling scheme to establish a bound $\boldsymbol{\beta}$ of 3 was a fraction (0.001%) that of [165] and 1% that of [185].

In the second study, we applied MARL with robustness against Byzantine adversaries using constrained coalition formation (CCF) games to achieve throughput maximisation of fog gateway and low-level controller's placement for co-offloading in wireless (SDN) hierarchical networks. Dfog-DRF scheme was proposed in **Chapter 5** using positive and negative feedback or pairwise-fairness (QoS) constraints to solve the *fog link scheduling with bottleneck fairness* problem. By devising a transformation process framework for fairness constraints applying unsupervised learning, the scheduling scheme was able using a fast-recursive policy control mechanism, to determine efficient fog gateway and low-level controller's placement (joint) policy or strategy profile. This centralised policy control mechanism for priority scheduling was benchmarked against the ***asset fair policy*** – a mixed integer programming (MIP) implementation of the problem which shows superior performance in complexity analysis (only in terms of speed). For 20 and 30 fog (router) nodes, the deterministic policy control mechanism takes a fraction of the time (< 1%) for both n=100 and n=1000 pairwise-fairness constraints. The proposed Dfog-DRF scheduling scheme also ensured throughput maximisation in joint multi-resource and multitask forwarding as compared to the distributed DRF scheduling scheme.

To evaluate empirically, the proposed research methods, the design of experimental testbeds used CloudSim and iFogSim. Discussed in **Chapter 6**, the CloudSim model and simulation kernel was briefed as well as simplistic extensions to CloudSim (e.g., Datacenter Broker class).

**Chapter 6** discussed in detail the experiment settings and results from comparative and empirical analysis.

**Overall objectives revisited.** Key (design and performance) objectives of both pieces of study were addressed in detail in **Chapters 4, 5 and 6**.

- **Architecture:** ***In Chapters 4 and 5***, centralised optimisation or fully decentralised learning techniques do not satisfy all essential fair allocation characteristics for SDN management of FEC. In such problems, hybrid scheduling scheme (centralised policy control applying distributed node offloading strategy feedback) where multiple administrative domains and logically distributed SDN controllers are used.
- **Load balancing for performance isolation and truthfulness:** These objectives were addressed in **chapters 4 and 5** in which scheduler manipulation, load balancing for performance isolation and Pareto optimisation using lexicographic method were important design goals. In the theoretical frameworks, hybrid scheduling mechanisms are designed to provide incentive compatibility and balance task allocation to controllers in specific cases to circumvent congestion and load imbalance between the distributed controllers.
- **Computational complexity: In chapter 6,** we compared the speed of the run-time optimisation algorithm with state-of-the-art techniques.
- **Resource utilisation: In chapter 6**, experiments showed the effect of load balancing on resource utilisation optimisation measured using the Bandwidth utilisation ratio: a measure of the network's topology and resources and reflection of the load condition of its links.
- **Quality of service**. Load balancing and strategy proof in distributed SDN to compare QoS guarantees of fog and edge networks are discussed **in chapter 6:**
    - ✓ Latency: the time taken by a packet to be transmitted from the source to the destination throughout the fog or edge network.

- ✓ Response time: The interval of time between request acquisition and acceptance until the request response - measured SDN's ability to meet QoS requirements in terms of availability, reactivity, and delay.
- ✓ Costs: the total cost of QoS-related efforts and deficiencies in joint offloading.
- ✓ Throughput: throughput was calculated as the processing speed of the network's nodes and its performance (measured as throughput = 1/ response time).

The research work presented in this thesis is appraised by revisiting the research questions and research objectives as formulated in **Chapter 1.**

**Study I.** How can monotonic scheduling be achieved for joint multi-resource and multitask offloading at the flat SDN-based federated edge?

Fair allocation in joint multi-resource and multitask offloading at the federated edge is behaviourally complex and this was adequately captured using coalition games with externalities (**see Chapters 2, 3, 4 and 6**). In Chapter 2, the research gap in fair (monotonic) scheduling for joint offloading at the Federated edge was discussed, Chapter 3 provided the method for rejuvenation of decentralised controller deployment from non-monotonicity in joint offloading due to positive and negative Controller-to-controller bargain externalities. Finally, Chapter 4 described the hybrid resource allocation mechanism in detail.

All objectives raised for Study I was addressed in Chapters 4 and 6. Chapter 4 addressed **design objectives** by proposing a reference architecture for SDN-based Federated edge. The core objectives were addressed via the proposed hybrid monotonic scheduling model for joint offloading as well as the algorithmic design of allocation policy optimisation. Chapter 6 provided empirical evidence, which covers objectives of computational complexity, resource utilisation and QoS attributes.

**Study II.** How can fog link scheduling with bottleneck fairness be achieved for joint multi-resource and multitask forwarding in dynamic wireless hierarchical SDN networks?

We addressed truthful scheduling with bottleneck fairness in joint forwarding via cross-layer throughput maximisation of fog gateway and low-level controller's placement in wireless hierarchical fog networks considering QoS constraints. The dynamicity of controller node strategies was captured as positive and negative feedback in our proposed solution applying constrained coalition formation games (**see Chapters 2, 3, 5 and 6**). In Chapter 2, the gap in fair link scheduling with bottleneck fairness for dynamic wireless ad-hoc networks was discussed. Chapter 3 introduced our constraint dimensionality reduction methodology for throughput maximisation of gateway and low-level controller's placement in wireless hierarchical fogs. The proposed fog link scheduling mechanism for joint multi-resource and multitask forwarding was detailed in Chapter 5.

For **Study II**, the core objectives raised for this study are answered in **Chapters 5 and 6**. In Chapter 5, the ***design objectives*** were addressed by first proposing an interference-aware link scheduling model for decentralised control applying CCF games and ground-truth propositional logic defined over a Boolean language. Next, the decentralised control learning methodology was developed applying positive and negative fairness constraints. The research objectives are further addressed by proposing a *map-reduce* framework to generate isometric clustering sets. A recursive MARL algorithm to adapt Pareto improvements in gateway and low-level controller's placement (allocation) policies was proposed in chapter 5. Chapter 6 discussed the building blocks of the experimental testbed using iFogSim and more importantly the results from simulated experiments to appraise performance objectives - resource utilisation and QoS.

## 7.2 Contributions revisited

Overall, the original contributions to literature are discussed **in Chapters 4, 5, 6.**

In the first study, the main contributions are listed as:

- This work proposed a scalable and robust control learning framework for multi-controller SDN applying the concept of software cybernetics and homeostasis for max-min fair scheduling of joint multi-resource and multi-task offloading at the SDN-based federated edge.

- We applied a reference architecture of the SDN-based federated edge modeled as ICGs and with controller-to-controller distributive bargain externalities (negative and positive) as cybernetic and homeostasis principles of feedback. A hybrid monotonic scheduling model was proposed as logically centralised policy control applying physically distributed node offloading feedback.
- We proposed the rejuvenation of decentralised controller deployment (East-West interface) from MARL policy for efficient joint computation offloading at the SDN-based federated edge architecture. The scheme addressed adversarial adjacent controllers caused by interactions that are sparse or direct and indirect network effects from parallel integrative and distributive EC bargain strategy.
- We developed IP-DRF, a generalisation of both integer partition and DRF (and DRFH) algorithms to guarantee monotonicity (population and resource) in the flat SDN control plane by evolving deterministic joint DRF policies guaranteed to be second-best Pareto-optimal (SBPO).
- Empirical results evaluated the efficiency of the generalised max-min fairness methodology to guarantee second best-Pareto optimality via simulation.

In the second study, the notable contributions to literature are as follows:

- This work proposed a fault-tolerant control learning framework for multi-controller SDN using software cybernetics and homeostasis to provide max-min fair scheduling of joint multi-resource and multi-task forwarding in wireless hierarchical fog networks.
- We applied a reference architecture modelled as co-channel interference using propositional logic for feature selection in fog gateway and low-level controller's placement. The proactive architecture applied hierarchical node offloading strategy feedback to generate fairness (positive and negative co-forwarding) constraints in homeostasis for the fog scheduling problem with bottleneck fairness.
- Our proposed Dfog-DRF scheme applied MARL with robustness against adversarial low-level controllers to ensure bottleneck fairness via QoS constraint dimensionality reduction methodology using CCF games. Dfog-DRF scheme provided proactive

rejuvenation of decentralised hierarchical controller deployment via deterministic policy for efficient joint forwarding at the hierarchical SDN control plane of wireless fog networks.

- We proposed unsupervised learning via the application of an orthonormal extraction (Map-Reduce) framework to support fast search of fog gateway and low-level controller's placement (joint) policies (or strategy profiles). The logically centralised policy management framework assisted sorting and reduction of joint policy space via application of pairwise-fairness constraints from homeostatic feedback.
- We applied a generalised recursive MARL algorithm to implement Pareto-improvements in cross-layer throughput maximisation of fog gateway and low-level controller's placement policy.

## 7.3 Challenges to Implementation

- Expense and unavailability of real-world environments - To decrease the complexity and separate quality concerns, simulators for cloud computing are required for system testing. These tools enable performance analysts to analyse the behavior of FEC environments by focusing on quality issues (fair allocation) under different scenarios.
- Optimisation Space: In certain instances, the problem space for the centralised policy optimisation grows exponentially and as such it can be cumbersome (not easy) to simulate FEC environments with many controller nodes (in our experiments we typically did not exceed 7 FEC nodes) without the use of graphics processing units (GPUs).

## 7.4 Limitation of the Thesis

The limitations of this thesis and possible future work are discussed in the last 2 sub sections.

### 7.4.1 Monotonic scheduling in the federated edge

- **Mixed bargain strategy externalities**: This work does not consider deterministic policy mechanisms for joint offloading at the federated edge with mixed externalities. Whereby, the environment states could be a negative Controller-to-controller bargain externality in some cases and positive Controller-to-controller bargain externalities in others.
- In some cases, the second-best (SBPO) may be unattainable.
- The decentralised control methodology proposed in this study assumes all DRF clusters are feasible – this may not be the case for some scenarios.

### 7.4.2 Scheduling with bottleneck fairness for Wireless hierarchical Fogs

- This work did not consider concurrent integrative and distributive controller bargaining strategies at the hierarchical SDN control plane leading to SBPO from MARL policies.
- The determination of load and size constraints for the decentralised control problem is rather simplistic, as we assume homogenous architectures for distributed micro-data centres. This computation is far more complex for the SDN architecture comprising heterogenous edge servers.
- **Anytime optimisation**: In experiments performed, only complexity analysis and empirical evaluation of the rejuvenation of decentralised controller deployment for joint forwarding was performed, a ratio bound (for solution accuracy) needs to be established to guarantee its applicability as an anytime solution.

## 7.5 Future Work in FEC

For Fair offloading at the federated edge with market failures, several directions can be explored for future work, these may include but are not limited to:

- Future work should consider fair resource allocation mechanisms for the federated edge with ***mixed bargain externalities.*** Whereby, an environment state could be a negative Controller-to-controller bargain externality in some cases and positive Controller-to-controller bargain externality in others.
- As mention earlier, the second-best (SBPO) allocation policy may be unattainable, in such cases the third-best solution [188] can be further explored for fair allocation.

In the second study, the following research work would be considered in the future.

- The determination of pairwise-fairness (positive and negative) or DRF constraints (load and size) is simplistic and fast because of the homogenous nature of multi-controllers in experiments, however, this computation is a lot more complex to establish fairness constraints for heterogenous data centre architectures in the control plane.
- Hybrid architectures may also be proposed leading to a more complex environment or the notion of population and resource non-monotonic environments.

# Bibliography


[1] P. Corcoran and S. K. Datta, “Mobile-Edge Computing and the Internet of Things for Consumers: Extending cloud computing and services to the edge of the network.,” *IEEE Consumer Electronics Magazine,* vol. 5, no. 4, pp. 73-74, 2016.

[2] Ma, Hua Dong, “Internet of things: Objectives and scientific challenges,” *Journal of Computer Science and Technology,* vol. 6, no. 26, pp. 991-924, 2011.

[3] R. Buyya and S. Srirama, Fog and Edge Computing: Principles and Paradigms, John Wiley & Sons., 2019.

[4] R. Jain and S. Paul, “Network virtualization and software defined networking for cloud computing: A survey,” *IEEE Communications Magazine,* vol. 51, no. 11, pp. 24-31, 2013.

[5] C. J. Bernardos, A. De La Oliva, P. Serrano, A. Banchs, L. M. Contreras, H. Jin and J. C. Zúñiga, “An architecture for software defined wireless networking,” *IEEE Wireless Communications,* vol. 21, no. 3, pp. 52-61, 2014.

[6] A. Ghodsi, M. Zaharia, B. Hindman, A. Konwinski, S. Shenker and I. Stoica, “Dominant resource fairness: Fair allocation of multiple resource types,” in *Proceedings of NSDI 2011: 8th USENIX Symposium on Networked Systems Design and Implementation*, Boston, 2011.

[7] W. Shi, J. Cao, Q. Zhang, Y. Li and L. Xu, “Edge Computing: Vision and Challenges,” *IEEE Internet of Things Journal,* vol. 3, no. 5, pp. 637-646, 2016.

[8] ITU-T-Rec-G.1000, *Communications Quality of Service: A Framework and Definitions,* 2001.

[9] N. Alhakbani, M. M. Hassan, M. A. Hossain and M. Alnuem, “A framework of adaptive interaction support in cloud-based internet of things (IoT) environment,” in *International conference on internet and distributed computing systems*, 2014.

[10] L. Li, S. Li and S. Zhao, “QoS-Aware scheduling of services-oriented internet of things,” *IEEE Transactions on Industrial Informatics,* vol. 10, no. 2, pp. 1497-1505, 2014.

[11] M.-a. Nef, L. Perlepes and G. I. Stamoulis, “Enabling QoS in the Internet of Things,” in *CTRQ 2012 : The Fifth International Conference on Communication Theory, Reliability, and Quality of Service*, 2012.

[12] B. Horling and V. Lesser, “A survey of multi-agent organizational paradigms,” *Knowledge Engineering Review,* vol. 19, no. 4, pp. 281-316, 2004.

[13] T. Rahwan, S. D. Ramchurn, V. D. Dang, a. Giovannucci and N. R. Jennings, “Anytime optimal coalition struture generation,” in *Proc. of the 22th National Conference on Artificial Intelligence*, 2007.

[14] G. Fioretti and B. Visser, “A cognitive interpretation of organizational complexity,” in *E:CO Emergence: Complexity and Organization*, Boston, Springer, 2006, pp. 495-513.

[15] J. W. Meek, J. De Ladurantey and W. H. Newell, “Complex systems, governance and policy administration consequences,” *E:CO Emergence: Complexity and Organization,* vol. 9, no. 1/2, p. 24, 2007.

[16] J. Foster, “From simplistic to complex systems in economics,” *Cambridge Journal of Economics,* vol. 26, no. 6, pp. 873-892, 2005.

[17] K. Sherif and B. Xing, “Adaptive processes for knowledge creation in complex systems: The case of a global IT consulting firm,” *Information and Management,* vol. 43, no. 4, pp. 530-540, 2006.

[18] F. Amagoh, “Perspectives on organizational change: systems and complexity theories,” *The Innovation Journal: The Public Sector Innovation Journal,* vol. 13, no. 3, pp. 1-14, 2008.

[19] L. A. Montuori, “Organizational longevity: Integrating systems thinking, learning and conceptual complexity,” *Journal of Organizational Change Management,* 2000.

[20] A. Ksentini, M. Bagaa, T. Taleb and I. Balasingham, “On using bargaining game for optimal placement of SDN controllers.,” in *IEEE International Conference on Communications (ICC)* , 2016.

[21] O. Blial, M. Ben Mamoun and R. Benaini, An Overview on SDN Architectures with Multiple Controllers, 2016.

[22] A. S. Tanenbaum and M. Van Steen, Distributed Systems: Principles and Paradigms, 2/E, Prentice-hall, 2007.

[23] K. Lancaster and R. G. Lipsey, “McManus on Second Best,” in *Readings in Welfare Economics*, London, Palgrave, 1973, pp. 213-214.

[24] I. E. Hafalir, “Efficiency in coalition games with externalities,” *Games and Economic Behavior,* vol. 61, no. 2, pp. 242-258, 2007.

[25] R. Guesnerie, “General statements on second best Pareto optimality,” *Journal of Mathematical Economics,* vol. 6, no. 2, pp. 169-194, 1979.

[26] A. Koshibe, A. Baid and I. Seskar, “Towards distributed hierarchical SDN control plane,” in *International Science and Technology Conference (Modern Networking Technologies)(MoNeTeC)*, 2014.

[27] M. Iorga, L. Feldman, R. Barton, M. J. Martin, N. Goren and C. Mahmoudi, “Fog Computing Conceptual Model,” NIST, North Charleston, 2018.

[28] F. Bannour, S. Souihi and A. Mellouk, “Distributed SDN Control: Survey, Taxonomy, and Challenges,” *IEEE Communications Surveys and Tutorials,* vol. 20, no. 1, pp. 333-354, 2018.

[29] B. Heller, R. Sherwood and N. McKeown, “The controller placement problem,” *ACM SIGCOMM Computer Communication Review,* vol. 42, no. 4, pp. 473-478, 2012.

[30] D. F. Spulber, “The Second Best Core,” *International Economic Review,* pp. 623-631, 1989.

[31] C. Kothari, Research methodology: methods and techniques, New Age International, 2004.

[32] L. Hurwicz and S. Reiter, Designing economic mechanisms, Cambridge University Press, 2006, p. 356.

[33] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. De Rose and R. Buyya, "CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software - Practice and Experience,* vol. 41, no. 1, pp. 23-50, 2011.

[34] H. Gupta, A. Vahid Dastjerdi, S. K. Ghosh and R. Buyya, "iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog computing environments," *Software: Practice and Experience,* vol. 47, no. 9, pp. 1275-1296, 2017.

[35] R. Buyya, R. Calherious, J. Son, A. Dastjerdi and Y. Yoon, "Software-defined cloud computing: Architectural elements and open challenges," in *International conference on advan 2014 International conference on advances in computing, communications and informatics (ICACCI) (pp. 1-12)*, Delhi, 2014.

[36] S. S. Yau, D. Huang, L. Zhu and K. Y. Cai, "A software cybernetics approach to deploying and scheduling workflows in service-based systems," in *11th IEEE International Workshop on Future Trends of Distributed Computing Systems (FTDCS'07) (pp. 149-156)*, Sedona, 2007.

[37] C. Liu, C. Jiang, H. Hu, K. Y. Cai, D. Huang and S. S. Yau, "A control-based approach to balance services performance and security for adaptive service based systems (ASBS)," in *33rd Annual IEEE International Computer Software and Applications Conference (Vol. 2, pp. 473-478)*, Seattle, 2009.

[38] J. Liu, J. Zhou and R. Buyya, "Software Rejuvenation Based Fault Tolerance Scheme for Cloud Applications," in *In 2015 IEEE 8th International Conference on Cloud Computing (pp. 1115-1118)*, New York, 2015.

[39] D. Adami, L. Donatini, S. Giordano and M. Pagano, "A network control application enabling Software-Defined Quality of Service," in *IEEE International Conference on Communications (ICC) (pp. 6074-6079)*, London, 2015.

[40] K. Ravindran, "Software cybernetics to manage adaptation behavior of complex network systems," in *23rd International Conference on Computer Communication and Networks (ICCCN) (pp. 1-8)*, Shanghai, 2014.

[41] M. A. Santos, B. A. Nunes, K. Obraczka, T. Turletti, B. T. De Oliveira and C. B. Margi, "Decentralizing SDN's control plane," in *39th Annual IEEE Conference on Local Computer Networks (pp. 402-405).*, Edmonton, 2014.

[42] T. Koponen, M. Casado, N. Gude, J. Stribling, L. Poutievski, M. Zhu, R. Ramanathan, Y. Iwata, H. Inoue, T. Hama and S. Shenker, Onix: A distributed control platform for large-scale production networks, 2019.

[43] S. O. Aliyu, F. Chen, Y. He and H. Yang, "A game-theoretic based QoS-aware capacity management for real-time edgeiot applications," in *IEEE International Conference on Software Quality, Reliability and Security (QRS) (pp. 386-397)*, Prague, 2017.

[44] F. Botelho, A. Bessani, F. M. Ramos and P. Ferreira, "On the design of practical fault-tolerant SDN controllers," in *Third European workshop on software defined networks (pp. 73-78)*, Budapest, 2014.

[45] S. O. Aliyu, F. Chen and Y. He, "QoS-Aware Resource Management in SDN-Based InterClouds: A Software Cybernetics Perspective," in *Proceedings - 2017 IEEE International Conference on Software Quality, Reliability and Security Companion, QRS-C 2017*, Prague, 2017.

[46] M. F. Bari, S. R. Chowdhury, R. Ahmed and R. Boutaba, "PolicyCop: An autonomic QoS policy enforcement framework for software defined networks," in *IEEE SDN for Future Networks and Services (SDN4FNS) (pp. 1-7)*, Trento, 2013.

[47] C. Perera and A. V. Vasilakos, "A knowledge-based resource discovery for Internet of Things," *Knowledge-Based Systems,* vol. 109, pp. 122-136, 2016.

[48] H. Flores, X. Su, V. Kostakos, A. Y. Ding, P. Nurmi, S. Tarkoma, P. Hui and Y. Li, "Large-scale offloading in the Internet of Things," in *IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops) (pp. 479-484)*, Kona, 2017.

[49] H. Zhu, "Formal specification of evolutionary software agents," in *International Conference on Formal Engineering Methods (pp. 249-261)*, Berlin, 2002.

[50] NIST, "NIST Definition of Cloud Computing," 2016. [Online].

[51] D. Ionescu, "Autonomic computing: The path towards controlling cloud computing services," in *I3rd IEEE International Symposium on Logistics and Industrial Informatics (pp. 11-11)*, Budapest, 2011.

[52] R. Buyya, R. N. Calheiros and X. Li, "Autonomic Cloud computing: Open challenges and architectural elements," in *Third international conference on emerging applications of information technology (pp. 3-10).* , Kolkata, 2012.

[53] P. Mayer, A. Klarl, R. Hennicker, M. Puviani, F. Tiezzi, R. Pugliese, J. Keznikl and T. Bure, "The autonomic cloud: A vision of voluntary, Peer-2-Peer cloud computing," in *7th IEEE International Conference on Self-Adaptation and Self-Organizing Systems Workshops (pp. 89-94)*, Philadelphia, 2014.

[54] E. F. Coutinho, D. G. Gomes and J. N. De Souza, "An Autonomic Computing-based architecture for cloud computing elasticity," in *2015 Latin American Network Operations and Management Symposium (LANOMS) (pp. 111-112)*, João Pessoa, 2015.

[55] K. Konstanteli, T. Cucinotta, K. Psychas and T. A. Varvarigou, "Elastic admission control for federated cloud services," *IEEE Transactions on Cloud Computing,* vol. 2, no. 3, pp. 348-361, 2014.

[56] D. Ardagna, S. Casolari, M. Colajanni and B. Panicucci, "Dual time-scale distributed capacity allocation and load redirect algorithms for cloud systems," *Journal of Parallel and Distributed Computing,* vol. 72, no. 6, pp. 796-808, 2012.

[57] N. M. Al Sallami, "Load balancing in green cloud computation," in *Proceedings of the World Congress on Engineering(pp. 3-5)*, London, 2013.

[58] D. Ardagna, B. Panicucci, M. Trubian and L. Zhang, "Energy-aware autonomic resource allocation in multitier virtualized environments," *IEEE Transactions on Services Computing,* vol. 5, no. 1, pp. 2-19, 2012.

[59] S. Ali, S.-y. Jing and S. Kun, "Profit-Aware DVFS Enabled Resource Management of IaaS Cloud," *International Journal of Computer Science Issues,* vol. 10, no. 2, p. 237, 2013.

[60] D. C. Marinescu, Cloud Computing: Theory and Practice: Second Edition, Morgan Kaufmann, 2017.

[61] N. Grozev and R. Buyya, “Inter-Cloud architectures and application brokering: Taxonomy and survey,” *Software - Practice and Experience,* vol. 44, no. 3, pp. 369-390, 2014.

[62] A. N. Toosi, R. N. Calheiros and R. Buyya, “Interconnected cloud computing environments: Challenges, taxonomy, and survey,” *ACM Computing Surveys (CSUR),* vol. 47, no. 1, pp. 1-47, 2014.

[63] R. Buyya, R. Ranjan and R. N. Calheiros, “InterCloud: Utility-oriented federation of cloud computing environments for scaling of application services,” in *IInternational Conference on Algorithms and Architectures for Parallel Processing (pp. 13-31)*, Berlin, 2010.

[64] F. Jrad, J. Tao and A. Streit, “SLA based service brokering in intercloud environments,” in *CLOSER 2012 - Proceedings of the 2nd International Conference on Cloud Computing and Services Science pp.76-81*, Porto, 2012.

[65] B. Rochwerger, D. Breitgand, A. Epstein, D. Hadas, I. Loy, K. Nagin, J. Tordsson, C. Ragusa, M. Villari, S. Clayman, E. Levy, A. Maraschini, P. Massonet, H. Muñoz and G. Tofetti, “Reservoir - When one cloud is not enough,” *Computer,* vol. 44, no. 3, pp. 44-51, 2011.

[66] F. Moscato, R. Aversa, B. Di Martino, T. F. Fortiş and V. Munteanu, “An analysis of mOSAIC ontology for cloud resources annotation,” in *federated conference on computer science and information systems (FedCSIS) pp. 973-980*, Szczecin, 2011.

[67] G. Aceto, A. Botta, W. De Donato and A. Pescapè, “Cloud monitoring: A survey,” *Computer Networks,* vol. 57, no. 9, pp. 2093-2115, 2013.

[68] N. Thakur, *Performance Testing in Cloud : A pragmatic approach,* Scientific Technical Committee, 2010.

[69] S. K. Garg, S. Versteeg and R. Buyya, “SMICloud: A framework for comparing and ranking cloud services,” in *4th IEEE International Conference on Utility and Cloud Computing, (UCC) pp. 210-218*, Victoria, 2011.

[70] M. Rak, A. Cuomo and U. Villano, “Cost/performance evaluation for cloud applications using simulation,” in *Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (pp. 152-157).* , Hammamet, 2013.

[71] T. Aoyama and H. Sakai, “Inter-cloud computing,” *Business and Information Systems Engineering,* vol. 53, no. 3, pp. 171-175, 2011.

[72] M. Aazam and E. N. Huh, “Cloud broker service-oriented resource management model,” *Transactions on Emerging Telecommunications Technologies,* vol. 28, no. 2, p. 2937, 2017.

[73] A. Takefusa, H. Nakada, R. Takano, T. Kudoh and Y. Tanaka, “GridARS: A grid advanced resource management system framework for Intercloud,” in *IEEE Third International Conference on Cloud Computing Technology and Science (pp. 705-710)*, Athens, 2011.

[74] J. Son, D. Barreto, R. N. Calheiros and R. Buyya, “Automatic Provisioning of Intercloud Resources driven by Nonfunctional Requirements of Applications,” in *Encyclopedia of Cloud Computing*, Wiley, 2016, p. 446.

[75] R. Mehrotra, S. Srivastava, I. Banicescu and S. Abdelwahed, "Towards an autonomic performance management approach for a cloud broker environment using a decomposition-coordination based methodology," *Future Generation Computer Systems,* vol. 54, pp. 195-205, 2016.

[76] K. M. Sim, "Agent-based cloud computing," *IEEE Transactions on Services Computing,* vol. 5, no. 4, pp. 564-577, 2012.

[77] J. O. Gutierrez-Garcia and K. M. Sim, "Agent-based cloud service composition," *Applied Intelligence,* vol. 38, no. 3, pp. 436-464, 2013.

[78] K. M. Sim, "Agent-Based Interactions and Economic Encounters in an Intelligent InterCloud," *IEEE Transactions on Cloud Computing,* vol. 3, no. 3, pp. 358-371, 2015.

[79] B. Varghese, N. Wang, J. Li and D. S. Nikolopoulos, "Edge-as-a-Service: Towards Distributed Cloud Architectures," in *International Conference on Parallel Computing*, Bristol, 2017.

[80] N. Wang, B. Varghese, M. Matthaiou and D. S. Nikolopoulos, "ENORM: A Framework For Edge NOde Resource Management," *IEEE Transactions on Services Computing,* vol. 1, no. 1, p. 99, 2017.

[81] K. M. Sim, "Agent-Based Fog Computing: Gossiping, Reasoning, and Bargaining," *IEEE Letters of the Computer Society,* vol. 1, no. 2, pp. 21-24, 2018.

[82] K. M. Sim, "Agent-based approaches for intelligent intercloud resource allocation," *IEEE Transactions on Cloud Computing,* vol. 7, no. 2, pp. 442-455, 2019.

[83] K. M. Sim, "Intelligent Resource Management in Intercloud, Fog, and Edge: Tutorial and New Directions," *IEEE Transactions on Services Computing,* pp. 1-1, 2020.

[84] M. E. Frincu, N. M. Villegas, D. Petcu, H. A. Müller and R. Rouvoy, "Self-healing distributed scheduling platform," in *Proceedings - 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, (CCGrid) pp. 225-234*, Newport Beach, 2011.

[85] W. Wang, B. Li and B. Liang, "Dominant resource fairness in cloud computing systems with heterogeneous servers," in *Proceedings - IEEE INFOCOM (pp. 583-591)*, Toronto, 2014.

[86] A. Celesti, F. Tusa, M. Villari and A. Puliafito, "Intercloud: The future of cloud computing. Concepts and advantages," in *Cloud Computing: Methodology, Systems, and Applications*, CRC Press, 2017, pp. 167-193.

[87] L. Mashayekhy and D. Grosu, "A coalitional game-based mechanism for forming cloud federations," in *Proceedings - 2012 IEEE/ACM 5th International Conference on Utility and Cloud Computing, (UCC) pp. 223-227*, Chicago, 2012.

[88] L. Mashayekhy, M. M. Nejad and D. Grosu, "Cloud federations in the sky: Formation game and mechanism," *IEEE Transactions on Cloud Computing,* vol. 3, no. 1, pp. 14-27, 2015.

[89] K. M. Sim, "A Computationally Efficient Bargaining Mechanism for Fog Commerce," *IEEE Letters of the Computer Society,* vol. 2, no. 1, pp. 5-8, 2019.

[90] D. Bernstein, D. Vij and S. Diamond, “An intercloud cloud computing economy technology, governance, and market blueprints,” in *Proceedings - 2011 Annual SRII Global Conference, (SRII) pp. 293-299*, San Jose, 2011.

[91] C. H. Hong and B. Varghese, “Resource management in fog/Edge computing: A survey on architectures, infrastructure, and algorithms,” *ACM Computing Surveys,* vol. 52, no. 5, pp. 1-37, 2019.

[92] R. Lavi and C. Swamy, “Truthful and near-optimal mechanism design via linear programming,” *Proceedings - Annual IEEE Symposium on Foundations of Computer Science, FOCS,* vol. 58, no. 6, pp. 1-24, 2005.

[93] T. Tran and D. Pompili, “Joint task offloading and resource allocation for multi-server mobile-edge computing networks,” *EEE Transactions on Vehicular Technology,* vol. 68, no. 1, pp. 856-868., 2018.

[94] D. Grosu and A. T. Chronopoulos, “Noncooperative load balancing in distributed systems,” *Journal of Parallel and Distributed Computing,* vol. 65, no. 9, pp. 1022-1034, 2005.

[95] J. Zheng, Y. Cai, Y. Wu and X. Shen, “Dynamic computation offloading for mobile cloud computing: A stochastic game-theoretic approach,” *IEEE Transactions on Mobile Computing,* vol. 18, no. 4, pp. 771-786, 2019.

[96] X. Chen, “Decentralized computation offloading game for mobile cloud computing,” *IEEE Transactions on Parallel and Distributed Systems,* vol. 26, no. 4, pp. 974-983, 2015.

[97] S. Ranadheera, S. Maghsudi and E. Hossain, "Computation offloading and activation of mobile edge computing servers: A minority game," *IEEE Wireless Communications Letters,* vol. 7, no. 5, pp. 688-691, 2018.

[98] J. Xu, B. Palanisamy, H. Ludwig and Q. Wang, "Zenith: Utility-Aware Resource Allocation for Edge Computing," in *Proceedings - 2017 IEEE 1st International Conference on Edge Computing, (EDGE) pp. 47-54*, Honolulu, 2017.

[99] M. Deng, H. Tian and X. Lyu, "Adaptive sequential offloading game for multi-cell Mobile Edge Computing," in *23rd International Conference on Telecommunications, (ICT) pp. 1-5*, Thessaloniki, 2016.

[100] F. Bonomi, R. Milito, P. Natarajan and J. Zhu, Fog computing: A platform for internet of things and analytics, Springer, Cham, 2014, pp. 169-186.

[101] M. Chiang and T. Zhang, "Fog and IoT: An Overview of Research Opportunities," *IEEE Internet of things journal,* vol. 3, no. 6, pp. 854-864, 2016.

[102] J. Li, C. Blake, D. S. De Couto, H. I. Lee and R. Morris, "Capacity of ad hoc wireless networks," in *Proceedings of the Annual International Conference on Mobile Computing and Networking, (MOBICOM) pp. 61-69*, Rome, 2001.

[103] S. Sardellitti, G. Scutari and S. Barbarossa, "Joint optimization of radio and computational resources for multicell mobile-edge computing," *IEEE Transactions on Signal and Information Processing over Networks,* vol. 1, no. 2, pp. 89-103, 2015.

[104] I. F. Akyildiz and X. Wang, "A survey on wireless mesh networks," *IEEE Communications Magazine,* vol. 43, no. 9, pp. 23-30, 2005.

[105] M. Younis and K. Akkaya, "Strategies and techniques for node placement in wireless sensor networks: A survey," *Ad Hoc Networks,* vol. 6, no. 4, pp. 621-655, 2008.

[106] R. Chandra, L. Qiu, K. Jain and M. Mahdian, "Optimizing the placement of Internet TAPs in wireless neighborhood networks," in *Proceedings - International Conference on Network Protocols, (ICNP) pp. 271-282*, Berlin, 2004.

[107] Y. Bejerano, "Efficient integration of multihop wireless and wired networks with QoS constraints," *IEEE/ACM Transactions on Networking,* vol. 12, no. 6, pp. 1064-1078, 2004.

[108] B. Aoun, R. Boutaba, Y. Iraqi and G. Kenward, "Gateway placement optimization in wireless mesh networks with QoS constraints," *IEEE Journal on Selected Areas in Communications,* vol. 24, no. 11, pp. 2127-2136, 2006.

[109] S. Mnguni, A. M. Abu-Mahfouz, P. Mudali and M. O. Adigun, "A review of gateway placement algorithms on internet of things," in *icABCD 2019 - 2nd International Conference on Advances in Big Data, Computing and Data Communication Systems (pp. 1-6)*, Durban, 2019.

[110] F. Zeng and Z. Chan, "Load balancing placement of gateways in wireless mesh networks with QoS constraints," in *Proceedings of the 9th International Conference for Young Computer Scientists, (ICYCS) pp. 445-450*, Hunan, 2008.

[111] D. Benyamina, A. Hafid and M. Gendreau, "Optimal placement of gateways in multi-hop wireless mesh networks: A clustering-based approach," in *Proceedings - Conference on Local Computer Networks, LCN*, Zurich, 2009.

[112] T. Vanhatupa, M. Hännikäinen and T. D. Hämäläinen, "Genetic algorithm to optimize node placement and configuration for WLAN planning," in *Proceedings of 4th IEEE*

*Internatilonal Symposium on Wireless Communication Systems 2007, (ISWCS) pp. 612-616*, Trondheim, 2007.

[113] F. Xhafa, A. Barolli, C. Sánchez and L. Barolli, "A simulated annealing algorithm for router nodes placement problem in Wireless Mesh Networks," *Simulation Modelling Practice and Theory,* vol. 19, no. 10, pp. 2276-2284, 2011.

[114] S. C. Ho, "An iterated tabu search heuristic for the single source capacitated facility location problem," *Applied Soft Computing Journal,* vol. 27, pp. 169-178, 2015.

[115] C. C. Lin, P. T. Tseng, T. Y. Wu and D. J. Deng, "Social-aware dynamic router node placement in wireless mesh networks," *Wireless Networks,* vol. 22, no. 4, pp. 1235-1250, 2016.

[116] C. C. Lin, "Dynamic router node placement in wireless mesh networks: A PSO approach with constriction coefficient and its convergence analysis," *Information Sciences,* vol. 232, pp. 294-308, 2013.

[117] K. Liyanage, M. Ma and P. Chong, "Controller placement optimization in hierarchical distributed software defined vehicular networks," *Computer Networks,* vol. 135, pp. 226-239, 2018.

[118] W. Wenjia, L. Junzhou and Y. Ming, "Gateway placement optimization for load balancing in wireless mesh networks," in *13th International Conference on Computer Supported Cooperative Work in Design (pp. 408-413)*, Saintiago, 2009.

[119] F. Li, Y. Wang, X. Y. Li, A. Nusairat and Y. Wu, "Gateway placement for throughput optimization in wireless mesh networks," *Mobile Networks and Applications,* vol. 13, no. 1, pp. 198-211, 2008.

[120] S. Basagni, "Distributed clustering for ad hoc networks," in *Proceedings Fourth International Symposium on Parallel Architectures, Algorithms, and Networks, (I-SPAN'99) pp. 310-315*, Perth, 1999.

[121] W. Saad, Z. Han, M. Debbah, A. Hjørungnes and T. Başar, "Coalitional game theory for communication networks," *IEEE Signal Processing Magazine,* vol. 26, no. 5, pp. 77-97, 2009.

[122] R. Massin, C. J. Le Martret and P. Ciblat, "A Coalition Formation Game for Distributed Node Clustering in Mobile Ad Hoc Networks," *IEEE Transactions on Wireless Communications,* vol. 16, no. 6, pp. 3940-3952, 2017.

[123] V. Cardellini, V. Grassi, F. Lo Presti and M. Nardelli, "Poster: Distributed QoS-aware scheduling in storm," in *DEBS 2015 - Proceedings of the 9th ACM International Conference on Distributed Event-Based Systems (pp. 344-347)*, Oslo, 2015.

[124] S. Lin, I. Akyildiz, P. Wang and M. Luo, "QoS-aware adaptive routing in multi-layer hierarchical software defined networks: A reinforcement learning approach," in *IEEE International Conference on Services Computing (SCC), pp. 25-33*, San Francisco, 2016.

[125] K. Zhang, Y. Mao, S. Leng, S. Maharjan and Y. Zhang, "Optimal delay constrained offloading for vehicular edge computing networks," in *IEEE International Conference on Communications (pp. 1-6)*, Paris, 2017.

[126] B. Brandenburger, Adam, Nalebuff, "The right game: Use game theory to shape strategy," *Harvard Business Review,* vol. 76, pp. 57-71, 1995.

[127] J. Nash, “Non-Cooperative Games,” *The Annals of Mathematics,* vol. 5, no. 4, p. 2, 1951.

[128] L. S. Shapley, “A Value for n-Person Games,” *Contributions to the Theory of Games,* vol. 28, pp. 307-317, 1953.

[129] T. Rahwan, *Algorithms for coalition formation in multi-agent systems,* Southampton: University of Southampton (Thesis), 2007.

[130] J. Y. Halpern, “Beyond Nash equilibrium: Solution concepts for the 21st century,” in *Proceedings of the twenty-seventh ACM symposium on Principles of distributed computing, pp. 1-10*, New York, 2011.

[131] A. Nowé, P. Vrancx and Y. De Hauwere, “Game theory and multi-agent reinforcement learning,” in *Reinforcement Learning* , Berlin, Springer, 2012, pp. 441-470.

[132] Q. Zhu and J. C. Oh, “An approach to dominant resource fairness in distributed environment,” in *In International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems (pp. 141-150)*, Berlin, 2015.

[133] T. Rahwan, T. Michalak, M. Wooldridge and N. R. Jennings, “Anytime coalition structure generation in multi-agent systems with positive or negative externalities,” *Artificial Intelligence,* no. 186, pp. 95-122, 2012.

[134] T. Rahwan, T. Michalak, E. Elkind, P. Faliszewski, J. Sroka, M. Wooldridge and N. R. Jennings, “Constrained coalition formation,” in *Proceedings of the National Conference on Artificial Intelligence (Vol. 25, No. 1)*, San Francisco, 2011.

[135] H. Yang, F. Chen and S. Aliyu, “Modern software cybernetics: New trends,” *Journal of Systems and Software,* vol. 124, pp. 169-186, 2017.

[136] K. Y. Cai, “Optimal software testing and adaptive software testing in the context of software cybernetics,” *Information and Software Technology,* vol. 44, no. 14, pp. 841-855, 2002.

[137] H. Hu, C. H. Jiang and K. Y. Cai, “Adaptive software testing in the context of an improved controlled Markov chain model,” in *32nd Annual IEEE International Computer Software and Applications Conference (pp. 853-858)*, Turku, 2008.

[138] B. Gaudin and A. Bagnato, “Software maintenance through Supervisory Control,” in *IEEE 34th Software Engineering Workshop, (SEW) pp. 97-105*, Limerick, 2012.

[139] J. M. Fernandes, S. Tjell, J. B. Jørgensen and Ó. Ribeiro, “Designing tool support for translating use cases and UML 2.0 sequence diagrams into a coloured petri net,” in *Proceedings - ICSE 2007 Workshops: Sixth International Workshop on Scenarios and State Machines, SCESM'07*, Minneapolis , 2007.

[140] Y. Yang and P. Gohari, “Embedded supervisory control of discrete-event systems,” in *IEEE International Conference on Automation Science and Engineering, pp. 410-415*, Edmonton, 2005.

[141] X. Y. Wang and K. Y. Cai, “Supervisory control of a kind of extended finite state machines,” in *24th Chinese Control and Decision Conference,(CCDC) pp. 775-780*, Taiyuan, 2012.

[142] D. Lorenzoli, L. Mariani and M. Pezzè, "Automatic generation of software behavioral models," in *Proceedings of the 30th international conference on Software engineering (pp. 501-510)*, Leipzig, 2008.

[143] X. Zhao, J. Xue, C. Hu, R. Ma and S. Zhang, "Research on software behavior modeling based on extended finite state automata," in *Communications security conference*, Beijing, 2014.

[144] Q. Yang, J. Lü, J. Xing, X. Tao, H. Hu and Y. Zou, "Fuzzy control-based software self-adaptation: A case study in mission critical systems," in *IEEE 35th Annual Computer Software and Applications Conference Workshops (pp. 13-18)*, Munich, 2011.

[145] Z. Ding, Z. Wei and H. Chen, "A software cybernetics approach to self-tuning performance of on-line transaction processing systems," *Journal of Systems and Software,* vol. 124, pp. 247-259, 2017.

[146] S. S. Yau, N. Ye, H. Sarjoughian and D. Huang, "Developing service-based software systems with QoS monitoring and adaptation," in *12th IEEE International Workshop on Future Trends of Distributed Computing Systems (pp. 74-80)*, Kunming, 2008.

[147] J. Park and K. Yeom, "A feedback-based approach to validate swrl rules for developing situation-Aware software," in *Proceedings - International Computer Software and Applications Conference*, Kyoto, 2013.

[148] V. Krishna and M. Perry, Efficient Mechanism Design, 2005.

[149] D. Vengerov, "A reinforcement learning approach to dynamic resource allocation," *Engineering Applications of Artificial Intelligence,* vol. 20, no. 3, pp. 383-390, 2007.

[150] R. S. Sutton and A. G. Barto, Reinforcement learning - second edition, Massachusetts: MIT press, 2018.

[151] J. Andrew Bagnell, S. Kakade, A. Y. Ng and J. Schneider, “Policy search by dynamic programming,” in *Advances in Neural Information Processing Systems*, 2004.

[152] S. Levine and V. Koltun, “Guided policy search,” in *International conference on machine learning (pp. 1-9)*, 2013.

[153] T. Eisenmann, G. Parker and M. W. Van Alstyne, “Strategies for two-sided markets,” *Harvard business review,* vol. 84, no. 10, p. 92, 2006.

[154] S. J. Grossman and O. D. Hart, “Takeover Bids, The Free-Rider Problem, and the Theory of the Corporation,” *The Bell Journal of Economics,* pp. 42-64, 1980.

[155] H. Moulin, “Fair Division in the Internet Age,” *Annual Review of Economics,* vol. 11, pp. 407-441, 2019.

[156] T. Michalak, J. Tyrowicz, P. McBurney and M. Wooldridge, “Exogenous coalition formation in the e-marketplace based on geographical proximity,” *Electronic Commerce Research and Applications,* 2009.

[157] N. Wiener, Cybernetics: or the Control and Communication in the Animal and the Machine, Paris: MIT Press, 1965.

[158] T. Rahwan, T. Michalak, N. R. Jennings, M. Wooldridge and P. McBurney, “Coalition structure generation in multi-agent systems with positive and negative externalities,” in *IJCAI International Joint Conference on Artificial Intelligence (pp. 257-263)*, Pasadena, 2009.

[159] L. Tassiulas and S. Sarkar, “Maxmin fair scheduling in wireless ad hoc networks,” *IEEE Journal on Selected Areas in Communications,* vol. 23, no. 1, pp. 163-173, 2005.

[160] L. Tassiulas and S. Sarkar, “Maxmin fair scheduling in wireless networks,” in *Twenty-First Annual Joint Conference of the IEEE Computer and Communications (INFOCOM) Societies (Vol. 2, pp. 763-772).*, New York, 2002.

[161] a. S. C. M. Misra, Sudip, Isaac Woungang, Guide to Wireless Ad Hoc Networks, Springer Science & Business Media, 2009.

[162] K. Holstein, J. W. Vaughan, H. Daumé, M. Dudík and H. Wallach, “Improving fairness in machine learning systems: What do industry practitioners need?,” in *In Proceedings of the 2019 CHI conference on human factors in computing systems (pp. 1-16)*, Glasgow, 2019.

[163] S. Sen and P. S. Dutta, “Searching for optimal coalition structures,” in *Proceedings - 4th International Conference on MultiAgent Systems, (ICMAS) pp. 287-292*, Paris, 2000.

[164] O. Shehory and S. Kraus, “Methods for task allocation via agent coalition formation,” *Artificial Intelligence,* vol. 101, no. 1-2, pp. 165-200, 1998.

[165] T. Sandholm, K. Larson, M. Andersson, O. Shehory and F. Tohmé, “Coalition structure generation with worst case guarantees,” *Artificial Intelligence,* vol. 111, no. 1-2, pp. 209-238, 1999.

[166] W. Ogryczak, H. Luss, M. Pioro, D. Nace and A. Tomaszewski, “Fair optimization and networks: A survey,” *Journal of Applied Mathematics,* no. Special, p. 25, 2014.

[167] R. Boadway and N. Bruce, "A general proposition on the design of a neutral business tax," *Journal of Public Economics,* vol. 24, no. 2, pp. 231-239, 1984.

[168] M. Satyanarayanan, P. Bahl, R. Cáceres and N. Davies, "The case for VM-based cloudlets in mobile computing," *IEEE Pervasive Computing,* vol. 8, no. 4, pp. 14-23, 2009.

[169] R. M. Thrall and W. F. Lucas, "N-person games in partition function form," *Naval Research Logistics Quarterly,* vol. 10, no. 1, pp. 281-298, 1963.

[170] J. C. Rochet and J. Tirole, "Platform competition in two-sided markets," *Journal of the european economic association,,* vol. 1, no. 4, pp. 990-1029, 2014.

[171] M. Satyanarayanan, "Pervasive computing: Vision and challenges," *IEEE Personal communications,* vol. 8, no. 4, pp. 10-17, 2001.

[172] K. W. Jee, D. L. McShan and B. A. Fraass, "Lexicographic ordering: Intuitive multicriteria optimization for IMRT," *Physics in Medicine and Biology,* vol. 52, no. 7, p. 1845, 2007.

[173] T. Rahwan and N. R. Jennings, "Distributing coalitional value calculations among cooperative agents," in *Proceedings of the National Conference on Artificial Intelligence (pp.152-157)*, Pittsburgh, 2005.

[174] S. O. Aliyu, F. Chen and H. Li, "A Self-Tuning Procedure for Resource Management in InterCloud Computing," in *IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C), pp. 326-333*, Vienna, 2016.

[175] T. Rahwan, T. P. Michalak, M. Wooldridge and N. R. Jennings, “Coalition structure generation: A survey,” *Artificial Intelligence,* vol. 229, pp. 139-174, 2015.

[176] S. Barocas, M. Hardt and A. Narayanan, “Fairness and Machine Learning,” *NIPs tutorial,* vol. 1, p. 2, 2018.

[177] C. Perera, Y. Qin, J. C. Estrella, S. Reiff-Marganiec and A. V. Vasilakos, “Fog computing for sustainable smart cities: A survey,” *ACM Computing Surveys,* vol. 50, no. 3, pp. 1-43, 2017.

[178] P. G. Naranjo, Z. Pooranian, M. Shojafar, M. Conti and R. Buyya, “FOCAN: A Fog-supported smart city network architecture for management of applications in the Internet of Everything environments,” *Journal of Parallel and Distributed Computing,* vol. 132, pp. 274-283, 2019.

[179] D. Magrin, M. Centenaro and L. Vangelista, “Performance evaluation of LoRa networks in a smart city scenario,” in *IEEE International Conference on Communications (ICC), pp. 1-7*, Paris, 2017.

[180] I. Gravalos, P. Makris, K. Christodoulopoulos and E. A. Varvarigos, “Efficient Network Planning for Internet of Things with QoS Constraints,” *IEEE Internet of Things Journal,* vol. 5, no. 5, pp. 3823-3836, 2018.

[181] R. Mitra and D. Nandy, “A Survey on Clustering Techniques for Wireless Sensor Network,” *International Journal of Research in Computer Science,* vol. 2, no. 4, pp. 51-57, 2012.

[182] R. Sepulchre, G. Drion and A. Franci, “Control Across Scales by Positive and Negative Feedback,” *Annual Review of Control, Robotics, and Autonomous Systems,* no. 2, pp. 89-113, 2019.

[183] J. Su, F. Lin, X. Zhou and X. Lu, “Steiner tree based optimal resource caching scheme in fog computing,” *China Communications,* vol. 12, no. 8, pp. 161-168, 2015.

[184] R. Buyya, R. Ranjan and R. N. Calheiros, “Modeling and simulation of scalable cloud computing environments and the cloudsim toolkit: Challenges and opportunities,” in *International conference on high performance computing & simulation (pp. 1-11)*, Leipzig, 2009.

[185] V. D. Dang and N. R. Jennings, “Generating coalition structures with finite bound from the optimal guarantees,” in *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS 2004*, New York, 2004.

[186] N. Ohta, “Coalition structure generation utilizing compact characteristic function representations,” in *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS (pp. 623-638)*, Lisbon, 2009.

[187] T. Rahwan, S. D. Ramchurn, V. D. Dang and N. R. Jennings, “Near-optimal anytime coalition structure generation,” in *IJCAI International Joint Conference on Artificial Intelligence (pp. 2365-2371)*, Lisbon, 2006.

[188] Y. K. Ng, “Theory of Third Best: How to Interpret and Apply,” *Pacific Economic Review,* vol. 22, no. 2, pp. 178-188, 2017.

# Appendix A

## A.1 Dominant Resource fairness: Multi-Resource Allocation

ALGORITHM 1: DRF Scheduler Pseudo-Code

$R = \langle r_1, \cdots, r_m \rangle$ ◃ total resource capacities

$C = \langle c_1, \cdots, c_m \rangle$ ◃ consumed resources, initially 0

$s_i$ $(i = 1 \ldots n)$ ◃ user i's dominant shares, initially 0

$Ui = \langle u_{i,1}, \cdots, u_{i,m} \rangle$ $(i = 1.n)$ ◃ resources given to user i, initially 0

**Pick** user i with lowest dominant share si

$D_i \leftarrow$ demand of user i's next task

If $(C+D_i) \leq R$ then

$C = C + D_i$ ◃ update consumed vector

$U_i = U_i + D_i$ ◃ update i's allocation vector

$s_i = \max^{m}_{j=1}\{u_{i,j}/r_j\}$

else return ◃ cluster is full

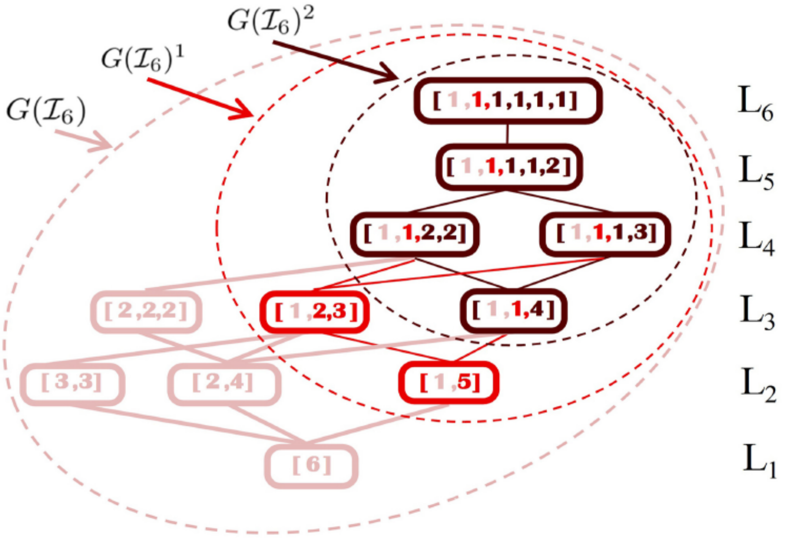
end if

## A.2 IP Algorithm and Pruning method

**Algorithm 1** Lowering the ratio bound with further search.

**if** $PFG^-$ **then**
Search $\mathcal{P}^A_{[n]}, \mathcal{P}^A_{[n-1,1]}, \mathcal{P}^A_{[n-2,1,1]}, \ldots, \mathcal{P}^A_{[1,1,\ldots,1]}$.
**else**
Search $\mathcal{P}^A_I$: $|I| \leqslant 2$ and then search $\mathcal{P}^A_{[n-2,1,1]}$.
**end if**
**for** $I \in \mathcal{I}^n$ **such that** $searched(\mathcal{P}^A_I) = 0$ **do**
$initialize(partition_I)$. {see Algorithm 1.1}
**end for**
**for** $\beta = \lceil \frac{n}{2} \rceil$ **down to** 2 {main loop} **do**
$r \leftarrow 1; c \leftarrow 1$.
**for** $I \in \mathcal{I}^n$ **such that** $searched(\mathcal{P}^A_I) = 0$ **and** $|partition_I| = \beta$ **do**
$rows[r] \leftarrow partition_I; \quad r \leftarrow r + 1$.
**for** $I', I'' \in partition_I$ **do**
**if** $I' \uplus I'' \notin columns$ **then**
$columns[c] \leftarrow I' \uplus I''; \quad c \leftarrow c + 1$.
**end if**
**end for**
**end for**
**for** $r = 1$ **to** $|rows|$ **do**
**for** $c = 1$ **to** $|columns|$ **do**
**if** $\exists I', I'' \in rows[r] : I' \uplus I'' = columns[c]$ **then**
$matrix[r][c] \leftarrow 1$.
**else**
$matrix[r][c] \leftarrow 0$.
**end if**
**end for**
**end for**
**for** $c = 1$ **to** $|columns|$ **do**
**if** $PFG^-$ **then**
$subspace[c] \leftarrow \mathcal{P}^A_{columns[c] \uplus I'}: I' \in \mathcal{I}^{n-|columns[c]|}, I' = [1, \ldots, 1]$.
**else**
$subspace[c] \leftarrow \mathcal{P}^A_{columns[c] \uplus I'} : I' \in \mathcal{I}^{n-|columns[c]|}, I' = [n - |columns[c]|]$.
**end if**
**end for**
**for** $c = 1$ **to** $|columns|$ **do**
**if** $searched(\mathcal{P}^A_{subspace[c]}) = 1$ **then**
$cost[c] \leftarrow 0$
**else**
$cost[c] \leftarrow |subspace[c]|$
**end if**
**end for**
**for** $c = 1$ **to** $|columns| - 1$ **do**
**for** $c' = c + 1$ **to** $|columns|$ **do**
**if** $subspace[c] = subspace[c']$ **then**
**for** $r = 1$ **to** $|rows|$ **do**
**if** $matrix[r][c'] = 1$ **then**
$matrix[r][c'] \leftarrow 0; \; matrix[r][c] \leftarrow 1$
**end if**
**end for**
**end if**
**end for**
**end for**
$\lambda \leftarrow integerSolver(matrix, cost)$ {call the integer solver}
**for** $c = 1$ **to** $|columns|$ **do**
**if** $\lambda[c] = 1$ **then**
Search through $subspace[c]$.
**end if**
**end for**
**for** $I \in \mathcal{I}^n$ **such that** $searched(\mathcal{P}^A_I) = 0$ **do**
**for** $c = 1$ **to** $|columns|$ **do**
**if** $\lambda[c] = 1$ **and** $\exists I', I'' \in partition_I$ **such that** $I' \uplus I'' = columns[c]$ **then**
$partition_I \leftarrow partition_I \setminus \{I', I''\} \cup \{I' \uplus I''\}$
**end if**
**end for**
**end for**
**end for**
Search through the remaining coalition structures. {This is to drop $\beta$ from 2 to 1.}

Algorithm 2: Integer Partition algorithm [133]




A.2-1 IP-Based Subspace Pruning [133] [158]

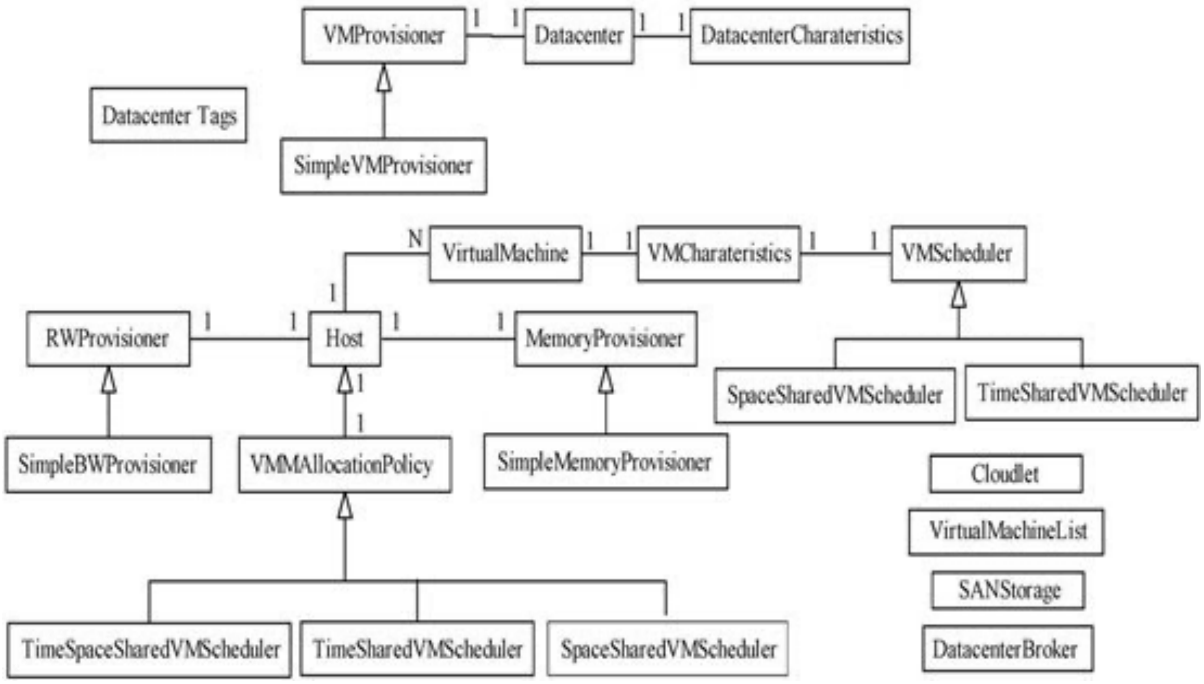



Figure A-0-1: CloudSim Class Diagram [184]

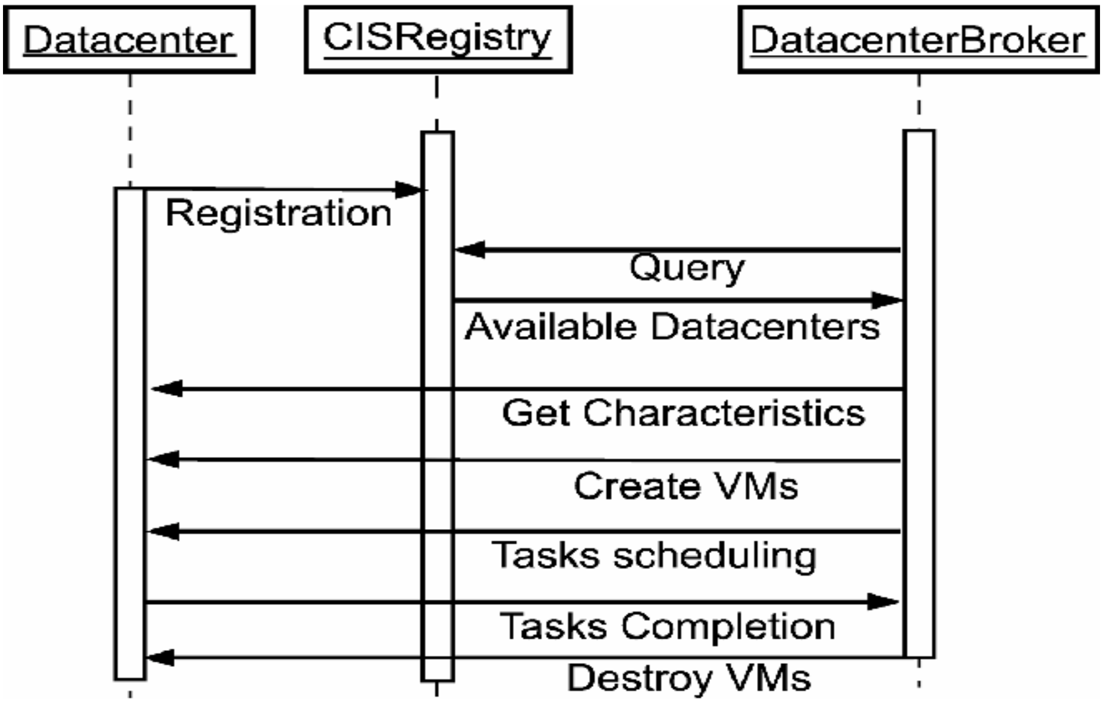



Figure A-0-2: CloudSim Communication flow [184]